

**UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS**

**FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

**E.A.P. DE INGENIERÍA DE SISTEMAS**

**Un Algoritmo GRASP-Reactivo para resolver el  
problema de cortes 1D**

**TESIS**

para optar el título de Ingeniero de Sistemas

**AUTOR**

Celso Ever Larico Mullisaca

**ASESOR**

David Mauricio

**Lima – Perú**

**2010**

**Celso Ever Larico Mullisaca**

**Un Algoritmo GRASP-Reactivo para Resolver el  
Problema de Cortes 1D**

“Tesis presentada a la Universidad Nacional Mayor  
de San Marcos, Lima, Perú, para obtener el Título  
de Ingeniero de Sistemas”

Orientador: Dr. David Mauricio.

UNMSM – LIMA

Septiembre 2010

Este trabajo está dedicado a toda mi familia  
en especial a mis padres.

Con todo mi corazón a Liria, la mujer que  
cambió mi vida.

## **AGRADECIMIENTOS**

A mi asesor David Mauricio Sánchez, por su orientación y dedicación brindada para que este trabajo cumpla con los objetivos trazados.

A mi familia por su apoyo a lo largo de mi carrera universitaria.

A mis amigos y a todas aquellas personas que indirectamente me ayudaron para culminar este trabajo con su apoyo invaluable.

Y, por encima de todo doy gracias a Dios.

# ÍNDICE

Lista de Figuras	xi
Lista de Tablas	xiii

## **CAPÍTULO 1: INTRODUCCIÓN ..... 1**

1.1	Antecedentes .....	1
1.2	El problema .....	1
1.3	Objetivos .....	3
1.3.1	General .....	3
1.3.2	Específico .....	3
1.4	Justificación .....	3
1.5	Propuesta .....	3
1.6	Alcance .....	4
1.7	Organización de la tesis .....	4

## **CAPÍTULO 2: MARCO TEÓRICO ..... 5**

2.1	Optimización Combinatoria .....	5
2.2	Heurísticas y Meta heurísticas .....	5
2.2.1	Heurísticas .....	5
2.2.2	Meta heurísticas .....	5
2.3	GRASP .....	7
2.3.1	GRASP Construcción .....	9
2.3.2	GRASP Búsqueda Local .....	10
2.3.3	Variantes del Algoritmo GRASP .....	10

## **CAPÍTULO 3: ESTADO DEL ARTE DEL PROBLEMA DE CORTES 1D..... 12**

3.1	Clasificación del Problema de Cortes .....	12
3.1.1	Dimensionalidad .....	12
3.1.2	Tipo de asignación .....	12
3.1.3	Agrupamiento de grandes objetos .....	12
3.1.4	Agrupamiento de piezas .....	12
3.2	Aplicaciones en la industria .....	13
3.2.1	Industria del plástico .....	13
3.2.2	Industria del aluminio .....	14
3.2.3	Industria de la construcción .....	14
3.2.4	Industria maderera .....	15
3.2.5	Industria del papel .....	15

3.2.6	Industria del cable .....	16
3.3	Métodos .....	16
3.3.1	Métodos Exactos .....	16
3.3.2	Métodos Heurísticos.....	17
3.3.3	Métodos Meta Heurísticos.....	17
3.4	Aplicativos .....	20
3.4.1	1DNest Versión 3.3 .....	20
3.4.2	1D Stock Cutter .....	21
3.4.3	CutLogic 1D .....	22
3.4.4	Real Cut 1D 7.8.5.0.....	23
<b>CAPÍTULO 4: UN ALGORITMO GRASP REACTIVO .....</b>		<b>25</b>
4.1	Evaluación y selección del método .....	25
4.2	Definición de variables y parámetros .....	26
4.3	Estructura de Datos .....	27
4.4	GRASP Reactivo.....	28
4.4.1	Algoritmo Principal.....	28
4.4.2	Algoritmo Construcción .....	29
4.4.3	Algoritmo Mejora.....	33
4.4.4	Algoritmo Registra_Mejor_Solucion .....	34
4.4.5	Descripción del algoritmo Calibrar .....	35
<b>CAPÍTULO 5: SISTEMA DE OPTIMIZACIÓN DE CORTE 1D .....</b>		<b>37</b>
5.1	Requerimientos mínimos de hardware y software.....	37
5.1.1	Configuración del hardware .....	37
5.1.2	Configuración del software .....	37
5.2	Descripción del Sistema .....	37
5.2.1	Estructura del Sistema .....	38
5.2.2	Módulos del Sistema .....	40
<b>CAPÍTULO 6: EXPERIMENTOS NUMÉRICOS.....</b>		<b>46</b>
6.1	Hardware y software .....	46
6.1.1	Hardware. ....	46
6.1.2	Software. ....	46
6.2	Instancias de prueba .....	46
6.3	Resultados Numéricos.....	48
6.4	Análisis de Resultados.....	54
6.4.1	Eficiencia Promedio .....	54
6.4.2	Eficiencia Ponderada.....	56
6.4.3	Convergencia.....	58

<b>CAPÍTULO 7: CONCLUSIONES Y TRABAJOS FUTUROS .....</b>	<b>60</b>
<b>ANEXO A: INSTANCIAS DE PRUEBA.....</b>	<b>63</b>

## Lista de figuras

1.1	Ejemplo de plan de corte 1D	2
2.1	Algoritmo GRASP	7
2.2	Criterio Goloso	9
2.3	Criterio GRASP	9
3.1	Rollos de Films transparentes	13
3.2	Tubos de Aluminio	14
3.3	Barras de Acero	14
3.4	Piezas de Madera	15
3.5	Rollos de Papel	15
3.6	Algoritmo GRASP para el problema de cortes 1D	18
3.7	Aplicativo 1D Nest Version 3.3	21
3.8	Aplicativo 1D Stock Cutter	22
3.9	Aplicativo CutLogic 1D	23
3.10	Aplicativo Real Cut 1D 7.8.5.0	24
4.1	Algoritmo GRASP Reactivo para el problema de cortes 1D	28
4.2	Algoritmo GRASP Construcción BFD para el problema de cortes 1D	30
4.3	Función que genera la LRC	31
4.4	Función que genera la LRC_B	32
4.5	Función eliminar resultado	32
4.6	Función Mejora	34
4.7	Función Mejor Solución	35
4.8	Función Calibrar parámetros de relajación	36
5.1	Módulos del sistema GraspCorte	39



5.2	Ventana principal del sistema GraspCorte	39
5.3	Ventana de configuración de la barra estándar del sistema GraspCorte	40
5.4	Ventana de configuración del Grasp básico del sistema GraspCorte	40
5.5	Ventana de configuración del Grasp Reactivo del sistema GraspCorte	41
5.6	Ventana de carga manual de piezas del sistema GraspCorte	42
5.7	Ventana de carga automática de piezas del sistema GraspCorte	42
5.8	Resultado final del GRASP básico del sistema GraspCorte	43
5.9	Resultado final del GRASP Reactivo del sistema GraspCorte	43
5.10	Reporte del GRASP básico del sistema GraspCorte	44
5.11	Reporte del GRASP Reactivo del sistema GraspCorte	45
6.1	Gráfica del promedio de la mejor eficiencia de GRASP Reactivo	55
6.2	Gráfica del ponderado de la mejor eficiencia de GRASP Reactivo	57
6.3	Gráfica de la mejor convergencia de GRASP	59

## Lista de tablas

1.1	Requerimiento de piezas	2
6.1	Lista de Grupos de instancias de prueba	47
6.2	Número de barras obtenidas por los criterios Grasp Construcción para el Grupo 1	48
6.3	Número de barras obtenidas por los criterios Grasp Construcción para el Grupo 2	49
6.4	Número de barras obtenidas por los criterios Grasp Construcción para el Grupo 3	49
6.5	Número de barras obtenidas por los criterios Grasp Construcción para el Grupo 4	50
6.6	Número de barras obtenidas por los criterios Grasp Construcción para el Grupo 5 STADLER	51
6.7	Número de barras obtenidas por los criterios Grasp Construcción para el Grupo 6 PAUTA	51
6.8	Número de barras obtenidas por los criterios Grasp Construcción para el Grupo 7 DEGRAEVE	52
6.9	Número de barras obtenidas por los criterios Grasp Construcción para el Grupo 8 GOULIMIS	52
6.10	Número de barras obtenidas por los criterios Grasp Construcción para el Grupo 9 HINTERDING & KHAN	52
6.11	Número de barras obtenidas por los criterios Grasp Construcción para el Grupo 10 ALOISE & MACULAN	53
6.12	Tabla resumen de eficiencias promedio	54
6.13	Tabla resumen de eficiencias ponderadas	56
6.14	Tabla resumen de convergencia	58

# Un Algoritmo GRASP-Reactivo para Resolver el Problema de Cortes 1D

## RESUMEN

Se tiene un grupo de requerimientos de piezas con una cantidad ilimitada de barras de algún tipo de material de tamaño estándar y éste posee mayor dimensión que el grupo de requerimientos. El problema de cortes 1D describe la utilización de las barras de tamaño estándar realizando cortes sobre ellas, de manera que se satisfaga todos los requerimientos con el menor número de barras de tamaño estándar. El problema es catalogado como NP-Difícil [Garey+79], y es ampliamente aplicado en diversos sectores de la industria tales como la maderera, vidrio, papelera, siderúrgica, etc.

La presente tesis propone dos algoritmos GRASP Reactivo para el problema de cortes 1D, basado en los algoritmos GRASP BFD y GRASP FFD propuestos por [Mauricio+02], además, desarrolla un sistema de optimización basado en los algoritmos propuesto. Se realizan experimentos numéricos del algoritmo propuesto sobre 100 instancias de pruebas, de donde se obtiene una eficiencia promedio de 97.04% y una eficiencia ponderada de 97,19% para el GRASP Reactivo BFD con proceso de mejoría, además se observa que el GRASP BFD con proceso de mejoría converge más rápido al encontrar una solución, donde realiza en promedio 1237 iteraciones. Los resultados numéricos muestran una mejora del GRASP Reactivo con respecto al GRASP básico implementado por Ganoza y Solano [Ganoza+02] que obtuvo una eficiencia promedio de 96.73%. Estas mejorías se pueden explicar porque el parámetro de relajación  $\alpha$  y  $\theta$  se ajusta de manera automática y es guiada en la búsqueda de una mejor solución.

Palabras clave: GRASP Reactivo, optimización combinatoria, meta heurísticas, problema de corte y empaquetado.

# A GRASP-Reactive Algorithm for Solving the Cutting Stock Problem 1D

## ABSTRACT

It has a set of requirements of parts with an unlimited number of bars of some kind of standard size and material and this has increased the group size requirements. The cutting stock problem 1D describes the use of standard-size bars of making cuts on them, so that it meets all requirements with the least number of standard size bars. The problem is listed as NP-Hard [Garey+79], and is widely used in various industry sectors such as wood, glass, paper, steel, and so on.

This thesis proposes two algorithms Reactive GRASP to the cutting stock problem 1D, based on the algorithms GRASP BFD and GRASP FFD proposed by [Mauricio+02], also, developed an optimization system based on the proposed algorithms.

Numerical experiments are conducted of the proposed algorithm on 100 instances of testing, where you get an average efficiency of 97.04% and a weighted efficiency of 97,04%, also be seen that the GRASP BFD with improvement converges faster to find a solution average of 1237 iterations. The numerical results show an improvement of reactive GRASP with respect to the basic GRASP implemented by Ganoza and Solano [Ganoza+02], who obtained an average efficiency of 96,73%. These improvements can be explained as the relaxation parameter  $\alpha$  and  $\theta$  is set automatically and is guided in the search for a better solution.

Keywords: Reactive GRASP, combinatorial optimization, metaheuristics, cutting stock problem.

## Capítulo 1: Introducción

### 1.1 Antecedentes

En los distintos procedimientos diarios en la industria, ocurren casos en los que debemos obtener pequeñas piezas con cierta dimensión a partir de barras de dimensión mayor. Ante esta situación, se seleccionará la manera de cortar y obtener las piezas pequeñas de las dimensiones requeridas, para ello se hará uso de las barras, de tal forma que el desperdicio se reduzca al mínimo.

El problema de cortes 1D ha sido estudiado desde años atrás de manera constante por diversos matemáticos y científicos, así se han publicado variedades de artículos, libros, revistas con un nivel especializado para que pueda ser puesta en práctica por empresas públicas y privadas, sin embargo, aún en la actualidad no ha sido adoptada totalmente [Vieira04].

No obstante, en estos últimos años y ante el creciente desarrollo de las grandes empresas, estas se han visto obligadas a ser más competitivas para poder afrontar el cambio, así han reducido costos y han aumentado los beneficios gracias a la utilización del modelo de soluciones de optimización que a lo largo de estos años se ha ido desarrollando, tal es el caso del denominado problema de cortes.

La primera formulación del problema de cortes fue realizada por el matemático Kantorovich en el año 1939 [Kantorovich39].

### 1.2 El problema

El Problema de Cortes 1D es generalmente conocido en la literatura como The Cutting Stock Problem [Dyckhoff90], y consiste en lo siguiente: dada una cantidad ilimitada de barras de longitud fija  $L$  y un conjunto de requerimientos (piezas) de dimensiones  $l_1, \dots, l_n$ , con  $l_i < L$ , se desea atender los requerimientos cortando cada barra  $L$ , de tal manera que utilicemos el menor número  $m$  de barras de longitud  $L$  con un mínimo desperdicio.

El problema presentado pertenece a uno más general, denominado “Problema de Cortes y Empaquetado” [Dyckhoff90].

Por ejemplo, tenemos una barra estándar  $L$ , cuyo tamaño es 130 metros y se requiere suplir los siguientes requerimientos dados en la Tabla 1.1, donde se describe el tamaño del requerimiento y la demanda que se necesita:

Tamaño	Demanda
40	1
20	4
50	2
100	1
80	1

Tabla 1.1. Requerimiento de piezas.

Una posible solución a esta atención de requerimiento sería la siguiente:

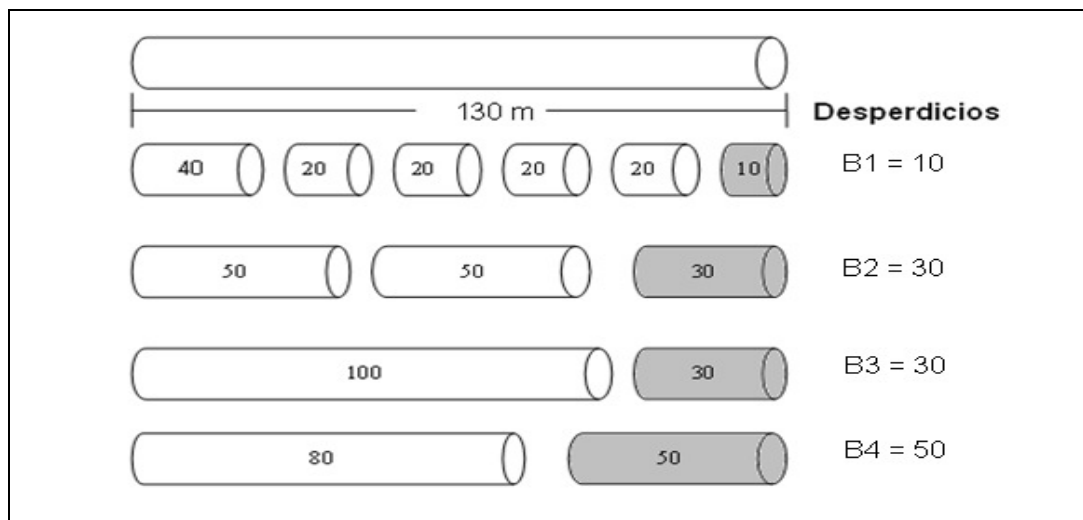


Figura 1.1: Ejemplo de plan de corte 1D.

De la Figura 1.1 observamos que se utilizaron cuatro barras estándar de longitud  $L = 130$  metros. El material sobrante total es de 120 metros, por lo que representa un índice de desperdicio de  $(120\text{m} / 4 \times 130\text{m}) \times 100 = 23\%$ .

## 1.3 Objetivos

### 1.3.1 General

Desarrollar un sistema para resolver el problema de cortes 1D a través de un algoritmo GRASP Reactivo, de tal manera que se reduzca aún más el desperdicio ocasionado en el proceso de corte.

### 1.3.2 Específico

- Estudiar y revisar el algoritmo GRASP para el problema de cortes 1D.
- Adaptar un algoritmo GRASP Reactivo para resolver el problema de cortes 1D.
- Evaluar la eficiencia de un algoritmo GRASP Reactivo respecto a un algoritmo GRASP.
- Desarrollar un software basado en el algoritmo GRASP Reactivo para optimizar el problema de cortes 1D.

## 1.4 Justificación

Se justifica la utilización de heurísticas por la naturaleza combinatoria del problema, al ser un problema NP-difícil [Garey+79], el costo computacional se eleva a medida que la cantidad de piezas a cortar de longitud diferente sean de gran tamaño.

El algoritmo GRASP ha tenido un desarrollo bastante fecundo para resolver problemas de optimización combinatoria.

Está comprobado que la variante reactiva da mejores resultados que el GRASP básico [Prais+98]. Entre tanto, en la literatura no se ha encontrado una aplicación del GRASP Reactivo para resolver el problema de cortes 1D.

En diversas industrias tales como la de metal, madera, plástico, papel, vidrio y aluminio, se evidencia muchas situaciones dentro de sus procesos que hace necesario resolver el problema de cortes 1D.

## 1.5 Propuesta

Introducir la calibración automática en el algoritmo GRASP para resolver el problema de cortes 1D.

## **1.6 Alcance**

En el presente trabajo se adapta una meta heurística GRASP reactiva para resolver el problema de cortes en una dimensión. Así mismo, se desarrolla un software que lo implemente, para ello se realizan pruebas numéricas con instancias tomadas de [Ganoza+02].

## **1.7 Organización de la tesis**

La presente tesis está organizada en siete capítulos; En el Capítulo 2 se describe el marco teórico y se explica brevemente los temas de optimización combinatoria, heurísticas y meta heurísticas, tipo de heurísticas y algoritmos meta heurísticos. En el Capítulo 3 se presenta el estado del arte, donde se describe la clasificación del problema de cortes, aplicaciones en la industria, métodos exactos, métodos heurísticos, métodos meta heurísticos y aplicativos (software) existentes en el mercado. En el Capítulo 4 se muestra el aporte teórico, donde se identifica un algoritmo GRASP Reactivo para el Problema de Cortes 1D. En el Capítulo 5 se describe la implementación de un Sistema de optimización de corte 1D, donde se implementa el GRASP básico y GRASP Reactivo; En el Capítulo 6 se muestran los experimentos numéricos realizados y los resultados obtenidos con el GRASP reactivo. Por último, en el Capítulo 7 se detalla las conclusiones y los trabajos futuros.



## Capítulo 2: Marco Teórico

En este capítulo se describen los conceptos relacionados a la resolución del problema de cortes, tales como optimización combinatoria, heurísticas y meta heurísticas, los tipos de heurísticas existentes, meta heurísticas existentes y la descripción de cada uno de ellos.

### 2.1 Optimización Combinatoria

La optimización combinatoria se define, como el “proceso encaminado a obtener el mejor resultado posible bajo un conjunto de circunstancias determinadas” [Campello+94].

Se define el conjunto base  $E = \{e_1, e_2, \dots, e_n\}$ , un conjunto de soluciones factibles  $F \subseteq 2^E$  y una función objetivo  $f: 2^E \rightarrow R$ . Se busca una solución óptima  $S^* \in F$  tal que  $f(S^*) \leq f(S), \forall S \in F$  [Resende03].

### 2.2 Heurísticas y Meta heurísticas

#### 2.2.1 Heurísticas

De acuerdo con ANSI/IEEE Std 100-1984 (American National Standards Institute / Institute of Electrical and Electronic Engineers) [IEEE1984], la heurística consiste en métodos o algoritmos exploratorios durante la resolución de problemas complejos en los cuales las soluciones se obtienen por la evaluación del progreso logrado en la búsqueda de un resultado final. No se puede garantizar que dichas soluciones sean las óptimas, pero sí razonablemente buenas.

#### 2.2.2 Meta heurísticas

Una meta heurística es un método heurístico para solucionar una clase muy general de problemas de cómputo donde se combinan los procedimientos de caja negra - generalmente heurísticos- de una manera eficiente. El nombre combina el prefijo griego “meta” (“más allá de”, aquí en el sentido de “a un nivel superior”) y “heurístico”.

La meta heurística se aplica generalmente a los problemas para los cuales no hay algoritmo específico satisfactorio o heurístico para su resolución; o cuando no es

práctico poner tal método en ejecución. La mayoría de las meta heurísticas de uso general se aplica sobre problemas de optimización combinatoria puros o todos aquellos que puedan adaptarse a esa naturaleza, como por ejemplo solucionar ecuaciones booleanas.

En [Osman+95] una definición dada es la siguiente: “Los procedimientos meta heurísticos son una clase de métodos aproximados que están diseñados para resolver problemas difíciles de optimización combinatoria, en los que los heurísticos clásicos no son ni efectivos ni eficientes. Las meta heurísticas proporcionan un marco general para crear nuevos algoritmos híbridos combinando diferentes conceptos derivados de la inteligencia artificial, la evolución biológica y mecanismos estadísticos”.

### **2.2.2.1 Búsqueda Tabú**

La búsqueda Tabú es un procedimiento meta heurístico de alto nivel introducido y desarrollado en su forma actual por [Glover89]. Su filosofía se basa en la explotación de diversas estrategias inteligentes para la resolución de problemas, basadas en procedimientos de aprendizaje [Glover+03]. A diferencia de otros algoritmos basados en técnicas aleatorias de búsqueda de soluciones cercanas, éste es determinista ya que elimina el azar en sus decisiones y la búsqueda del óptimo está guiada por una estrategia basada en el uso de estructuras de memoria, que guardan soluciones a corto y largo plazo.

### **2.2.2.2 Algoritmos Genéticos**

El Algoritmo Genético es un algoritmo de búsqueda que explora un espacio de solución que simula procesos en un sistema natural hacia la evolución, específicamente aquellos que siguen el principio de la supervivencia en función de la adaptabilidad [Adenso96]. Fue desarrollado por [Holland75] y se distingue muy claramente de todos los anteriores, básicamente por el hecho de que en cada iteración se tiene un conjunto de soluciones o población en curso y no una única solución en curso. Las soluciones sucesoras se obtienen a partir de parejas constituidas con los elementos de la población y no mediante la transformación de la solución en curso.

## 2.3 GRASP

La palabra GRASP son las siglas en inglés de *Greedy Randomize Adaptive Search Procedures*, que en español quiere decir *Procedimientos de Búsqueda basados en funciones o Procedimientos Golosos Aleatorios Adaptativos*. El desarrollo del método GRASP se dio a conocer en el trabajo de [Feo+95], el cual ha tenido un desarrollo satisfactorio para resolver problemas de optimización combinatoria [Resende03].

GRASP es un método multi-arranque procedimiento iterativo que consiste en una fase de construcción y una de mejora. En la fase de construcción se aplica un procedimiento heurístico constructivo para seleccionar iterativamente una buena solución inicial. Esta solución se mejora en la segunda fase mediante un algoritmo de búsqueda local. Luego de repetirse este procedimiento, la mejor de todas las soluciones examinadas se guarda como resultado final [Resende03].

### Procedimientos GRASP

El método GRASP, a diferencia del criterio usado por el algoritmo goloso, crea un conjunto de candidatos para encontrar una solución, de este modo seleccionará uno de ellos de forma aleatoria. En consecuencia, GRASP permite construir varias soluciones distintas.

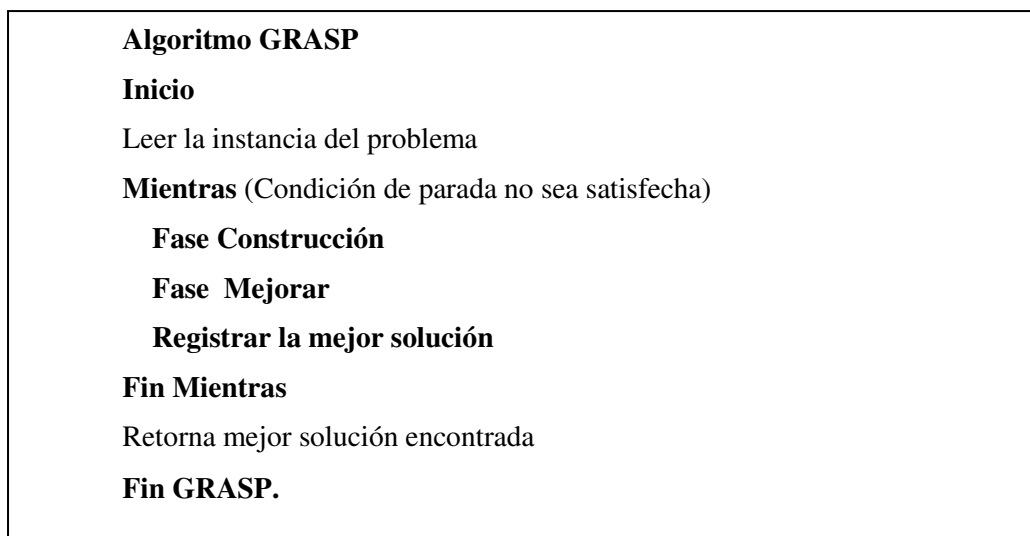


Figura 2.1. Algoritmo GRASP.

## **Fases del Algoritmo GRASP**

### ***a) Leer instancias del problema***

Se ingresan necesariamente los siguientes parámetros:

- Los datos o requerimientos que procesará el sistema.
- El parámetro de relajación  $\alpha$ .
- Los datos para el criterio de la condición de parada, tales como: la cantidad de iteraciones que realizará en total o el tiempo máximo de procesamiento, etc.

### ***b) Condición de Parada***

El GRASP genera una solución por cada iteración realizada, pero se debe indicar el fin de esta iteración mediante condiciones de paradas, éstas pueden ser:

- *Condición de Optimalidad.* Condición matemática que si es verificada para una solución, entonces se afirma que ésta es óptima.
- *Condición de Tiempo.* Se define un tiempo máximo de procesamiento del algoritmo y, una vez culminado ese tiempo, el algoritmo debe terminar y mostrar la mejor solución.
- *Condición de número de iteraciones.* Se define un número máximo de iteraciones que el algoritmo debe realizar y, una vez finalizada las iteraciones, el algoritmo debe terminar y presentar el mejor resultado hallado.
- *Híbrido.* Consiste en la combinación de dos o más condiciones de parada.

### ***c) Fase de Construcción***

Se construye una solución en forma iterativa. En cada iteración se selecciona una lista de elementos candidatos llamada Lista Restringida de los mejores Candidatos, luego se selecciona un elemento aleatorio de la Lista Restringida.

### ***d) Fase Mejorar***

Se realiza un proceso de búsqueda local a partir de la solución construida y se trata de mejorar la solución encontrada hasta que no se pueda mejorar más.

### ***e) Registrar la mejor solución***

En este último procedimiento se compara la mejor solución obtenida con la solución encontrada en la iteración actual, si la solución encontrada en la iteración actual es

mejor que la mejor solución entonces la nueva mejor solución será registrada, de lo contrario se mantiene la mejor solución. Se culmina con el registro cuando la iteración cumpla con la condición de parada. Al terminar la iteración se registra la mejor solución encontrada [Feo+95].

Por lo tanto, el método GRASP presenta dos principales fases: la primera fase se refiere a la construcción goloso - aleatorio de una solución, y la segunda fase a la mejora de la solución construida. La calidad de la solución encontrada dependerá del parámetro de relajación y de la condición de parada [Feo+95].

### 2.3.1 GRASP Construcción

En esta fase se construye una solución golosa y aleatoria, donde se emplea un criterio Goloso:

$$x^* := \text{Mejor}\{f(x) : x \in E\}$$

**Figura 2.2. Criterio Goloso.**

Donde:

$f$ : Es la función golosa.

$E$ : Conjunto de objetos.

$x^*$ : Elemento que presenta mejor valor de  $f$ .

El criterio Goloso dado en la Figura 2.2 dice que: el candidato que formará parte del conjunto solución es el elemento (variable u objeto) no tratado que presenta mejor (mayor o menor) valor para la función objetivo.

GRASP relaja el criterio goloso y construye un conjunto de posibles objetos que se denomina Lista Restringida de Candidatos (LRC) como se muestra en la Figura 2.3.

$$\begin{aligned} \bar{x} &:= \text{Mejor}\{f(x) : x \in E\}, \\ \underline{x} &:= \text{Peor}\{f(x) : x \in E\}, \\ LRC &:= \{x \in E : \bar{x} - \alpha(\bar{x} - \underline{x}) \leq f(x) \leq \bar{x}\}, \\ \text{Seleccionado} &:= \text{Random}(LRC); \end{aligned}$$

**Figura 2.3. Criterio GRASP.**

Para generar el conjunto LRC, se utiliza el parámetro  $\alpha$ , que se encarga de transformar la solución construida en aleatoria. Este criterio construye distintas soluciones en cada iteración haciendo uso de  $\alpha$ .

El valor del parámetro de relajación es seleccionado por el usuario, tomando valores que se encuentran entre 0 y 1. Observar que si  $\alpha = 0$  entonces el criterio de selección es un criterio goloso, mientras que si  $\alpha = 1$ , entonces es totalmente aleatorio. Así mismo, proporciona un intervalo permisible donde se puede encontrar soluciones óptimas [Ganoza+02].

El componente adaptativo de GRASP se muestra porque, en algunos casos, la función golosa puede variar en cada iteración de la fase de construcción, esto quiere decir que el valor asignado a cada una de las candidatas es actualizado en cada iteración.

Saliendo de la fase construcción se tiene una instancia Solución del problema, que puede ser mejorada [Ganoza+02].

### 2.3.2 GRASP Búsqueda Local

Como se observó con las heurísticas, las soluciones generadas por GRASP son cercanas a las óptimas, por lo que es necesaria la aplicación de un procedimiento de mejora de la solución encontrada en la fase de Construcción [Feo+95].

Se define una vecindad de soluciones que rodean la solución encontrada. Luego de construir la vecindad de soluciones, se realiza una búsqueda de la mejor solución entre la vecindad de soluciones y la solución que se va a mejorar, se comparan las soluciones y, en caso de que se mejore la solución, se reemplaza la solución y nuevamente se repite el proceso. El proceso termina cuando no se pueda realizar la mejora.

### 2.3.3 Variantes del Algoritmo GRASP

#### 2.3.3.1 GRASP Reactivo

El GRASP Reactivo permite el auto ajuste del parámetro  $\alpha$  en función de la calidad de soluciones previamente encontradas. En lugar de utilizar un valor fijo para el parámetro  $\alpha$ , se propone seleccionar un valor al azar de un conjunto discreto  $\Psi = \{\alpha_1, \dots, \alpha_h\}$  como un conjunto de valores posibles para el parámetro  $\alpha$ . El uso de valores diferentes de  $\alpha$  en diferentes iteraciones permite la construcción de diferentes Listas Restrictas de

Candidatos, conduciendo así a la construcción de diferentes soluciones las que nunca hubieran sido construidas si se usaba un valor fijo de  $\alpha$ . Tenemos inicialmente que todos los valores de  $\alpha_i$  poseen la misma probabilidad de ser elegido con una probabilidad  $P_i = 1/m$ ,  $i = 1, \dots, m$  correspondiente a una distribución uniforme. En el proceso iterativo del GRASP se guarda el valor promedio de las soluciones encontradas en  $A_i$ , por cada valor de  $\alpha_i$ . Después de cierto número de iteraciones las probabilidades

son reevaluadas siendo  $p_i = q_i / \sum_{j=1}^h q_j$ , con  $q_i = Z^* / A_i$  para  $i = 1, \dots, h$ , siendo  $Z^*$

la mejor solución encontrada hasta el momento. El valor de  $q_i$  será mayor para valores de  $\alpha = \alpha_i$  que proporcionen las mejores soluciones en promedio. Entonces para mayores valores de  $q_i$  corresponden a valores de  $\alpha$  más adecuados [Prais+98], [Resende03].

### 2.3.3.2 Path Relinking

Esta variante fue aplicada para un algoritmo GRASP por [Laguna+99] de la propuesta hecha originalmente por [Glover+97].

Dada la desventaja de GRASP al no poseer estructuras de memoria, se propone como alternativa de solución el uso de reencadenamiento de trayectorias, es decir, Path Relinking.

La idea de Path Relinking es buscar soluciones que compartan atributos con antiguas soluciones con la esperanza de obtener mejores soluciones.

## Capítulo 3:

# Estado del Arte del Problema de Cortes 1D

En el presente capítulo se detallará la clasificación del problema de cortes y se mostrará brevemente la ubicación del problema de cortes 1D, además se describirán las técnicas existentes en la literatura para el problema a tratar, las aplicaciones en las industrias, los aplicativos software que se utilizan y una comparación de las meta heurísticas.

### 3.1 Clasificación del Problema de Cortes

El problema de cortes se clasifica en base a cuatro características importantes, el cual ha sido propuesta por Dyckhoff [Dyckhoff90]. Estas características son: dimensionalidad, tipo de asignación y agrupamiento de objetos y de piezas.

#### 3.1.1 Dimensionalidad

(1) una-dimensión, (2) dos-dimensiones, (3) tres-dimensiones, (N) N-dimensiones con  $N > 3$ .

#### 3.1.2 Tipo de asignación

(B) Todos los objetos y una selección de ítems, (V) Una selección de objetos y todos los ítems.

#### 3.1.3 Agrupamiento de grandes objetos

(O) Un objeto, (I) Figuras idénticas, (D) Diferentes figuras.

#### 3.1.4 Agrupamiento de piezas

(F) Pocas piezas (de formas diferentes), (M) Muchas piezas de muchas formas diferentes, (R) Muchas piezas de relativamente pocas formas diferentes, (C) Figuras congruentes.

Dyckhoff hace uso de una notación de 4-tuplas representadas por los símbolos  $\alpha/\beta/\gamma/\delta$  [Dyckhoff90] de las cuatro características descritas arriba respectivamente, y de este modo representar el tipo de problema de cortes. Para el caso del problema de



cortes en 1D la notación respectiva tendría la siguiente forma:  $1 / V / I / R$ , que quiere denotar que el problema es de una dimensión (1), con una selección de objetos (barras a cortar) y todos los ítems(V), con agrupamiento de objetos de figuras idénticas (barras de igual longitud) (I), y con agrupamiento de muchas piezas de relativamente pocas formas distintas (R).

### 3.2 Aplicaciones en la industria

Los problemas de cortes 1D se presentan en una gran cantidad de situaciones. Entre ellas está la industria del metal [Karelahti02] para el corte de varillas o barras, la industria de la madera [Morabito+97], la industria del vidrio [Dyson76], la industria del plástico [Hatori98] y la industria del papel [Westernlund+95], donde se requieren cortar tubos de madera, metal, bobinas de papel, etc.

#### 3.2.1 Industria del plástico

Minimizar la merma en el corte de rollos para película (Films), esto dependerá además del tipo de material, de la longitud y cantidad de la orden.



**Figura 3.1. Rollos de Films transparentes.**

### 3.2.2 Industria del aluminio

Fábricas que usan como materia prima el aluminio. Este material se utiliza en construcción civil, por las características que posee, las cuales les ayuda a obtener ventajas sobre la competencia.



**Figura 3.2. Tubos de Aluminio.**

### 3.2.3 Industria de la construcción

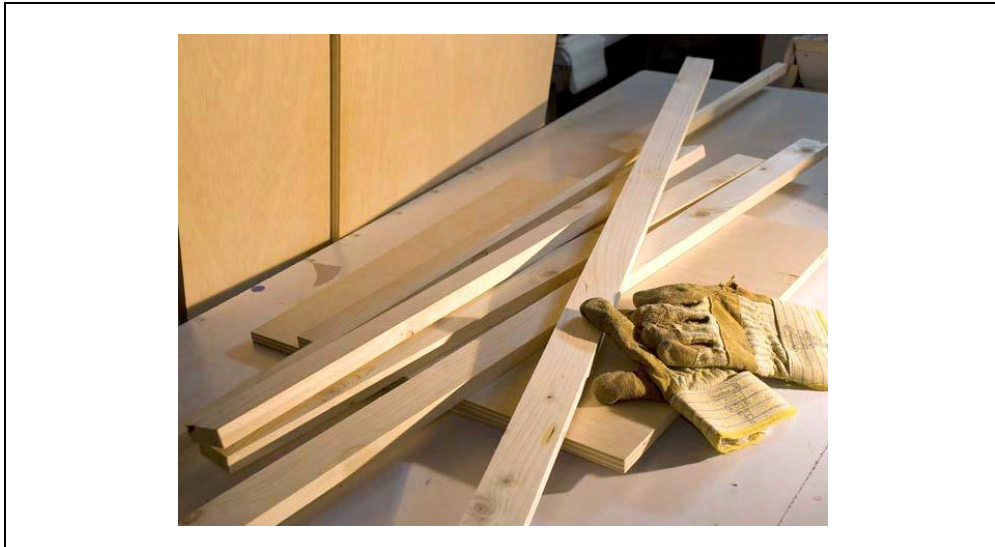
Industria de la Construcción. Minimizar la merma de las barras que se usa en la construcción de edificios, departamentos, puentes, etc., reduciendo los costos en un margen notorio.



**Figura 3.3. Barras de Acero.**

### 3.2.4 Industria maderera

Se aplica al momento de realizar los cortes estándar de listones de madera de un mismo ancho para la construcción de diferentes muebles.



**Figura 3.4. Piezas de Madera.**

### 3.2.5 Industria del papel

Los requerimientos que se satisfagan de los clientes cuando soliciten rollos de papel de diferente longitud. Por ejemplo, de papel higiénico o papel toalla.



**Figura 3.5. Rollos de Papel.**

### **3.2.6 Industria del cable**

Aquellas que se dedican a la venta de rollos de cable, alambre, etc., prefiriendo minimizar el sobrante en el corte de cable al realizar las ventas.

## **3.3 Métodos**

### **3.3.1 Métodos Exactos**

Los algoritmos que dan una solución exacta para la gran mayoría de problemas de cortes y empaquetados consumen un alto costo computacional, lo que inhibe su aplicación en situaciones concretas, incluso para los pequeños problemas [Vieira04]. Las técnicas exactas aplicadas están relacionadas con el modelo del problema de cortes 1D ya que el problema puede ser modelado como programa lineal entera. Los métodos lineales que se utilizan son tales como el de generación de columnas, y métodos de ramificación y acotación, programación dinámica y enumeración implícita, entre otros [Delgadillo07].

#### **3.3.1.1 Método de Ramificación y Acotación**

Propuesto por [Caprara98], es una técnica muy conocida para resolver problemas combinatorios, se basa en la reducción del espacio de estado del problema (conjunto factible) donde se podan las áreas en que no podrían producirse mejores resultados que los ya encontrados previamente. Este método busca un espacio finito de estados  $S$  y un  $S^*$  que es el resultado óptimo de la función objetivo. El método desarrolla un árbol donde cada nodo representa una parte del espacio de estado  $S$ .

#### **3.3.1.2 Método de Generación de Columnas**

Cuando el tamaño de los patrones de corte es muy grande, como los generados en los sectores industriales al determinar los posibles patrones de corte, puede resultar una tarea sumamente imposible [Delgadillo07].

Para abordar esta dificultad, una forma de resolver fue dada por Gilmore y Gomory [Gilmore+61], la propuesta consiste en trabajar con pocos patrones de corte a la vez en una matriz  $m \times m$  (base factible de dimensión igual al número de ítems diferentes) y generar nuevos patrones (nuevas columnas para la base) solamente cuando realmente se necesiten.

El método consiste en que a cada iteración del método Simplex (el paso del pivotamiento, variable entrante, es decir el patrón de corte) se resuelva un problema de la mochila asociada al problema.

Para una descripción completa y detallada de este método, en [Poldi03] se aborda este tema de manera completa.

### **3.3.2 Métodos Heurísticos**

Para la resolución del problema de mediano y de grande porte, se aplican los métodos heurísticos que se han desarrollado, muchos de ellos para alcanzar soluciones viables. Las heurísticas se agrupan de acuerdo al comportamiento del problema: corte en lotes y de tipo on-line.

#### **3.3.2.1 Corte en lotes**

Los cortes se realizan cuando se poseen todos los requerimientos de piezas a cortar. Entre las heurísticas existentes se tiene: BF (Best Fit), NFD (Next Fit Decreasing), FFD (First Fit Decreasing) y BFD (Best Fit Decreasing) [Coffman96].

La heurística FFD ordena previamente los ítems de manera decreciente, luego selecciona la primera barra ya utilizada de tal modo que encaje la pieza a cortar, si no es posible se utiliza una nueva barra.

La heurística BFD ordena previamente los ítems de manera decreciente, luego selecciona la barra ya utilizada de tal modo que minimice la longitud sobrante al encajar la pieza a cortar, si no es posible se utiliza una nueva barra.

#### **3.3.2.2 Corte on-line**

Los cortes de las piezas se llevan a cabo conforme llegan los requerimientos y no se espera a tener el total de éstos. Así se tienen las heurísticas: NF(Next Fit) y FF (First Fit) [Coffman96].

### **3.3.3 Métodos Meta Heurísticos**

Las meta heurísticas usan algunas de las heurísticas existentes combinándolas para obtener resultados cercanos a la óptima.

### 3.3.3.1 GRASP

A continuación se describirá el desarrollo de la meta heurística GRASP basado en el desarrollo de [Mauricio+02]:

#### GRASP para el problema de cortes 1D

Considerar la notación siguiente:

- $L$ : la longitud estándar de barras a utilizar
- $n$ : el número total de cortes requeridos
- $E(i)$ : el tamaño del requerimiento “ $i$ ”
- $i$ : el requerimiento, donde  $0 < i = n$

Asumiremos que la demanda de cada uno de los requerimientos es unitaria. En caso contrario, sólo bastará replicar el requerimiento las veces que sea necesaria la demanda.

El criterio utilizado para seleccionar la barra que atenderá a un requerimiento de tamaño  $E(i)$  es el Goloso FFD.

Las partes principales del algoritmo desarrollado son las siguientes:

- GRASP Construcción.
- Mejor Solución.

El algoritmo descrito de manera general es como sigue:

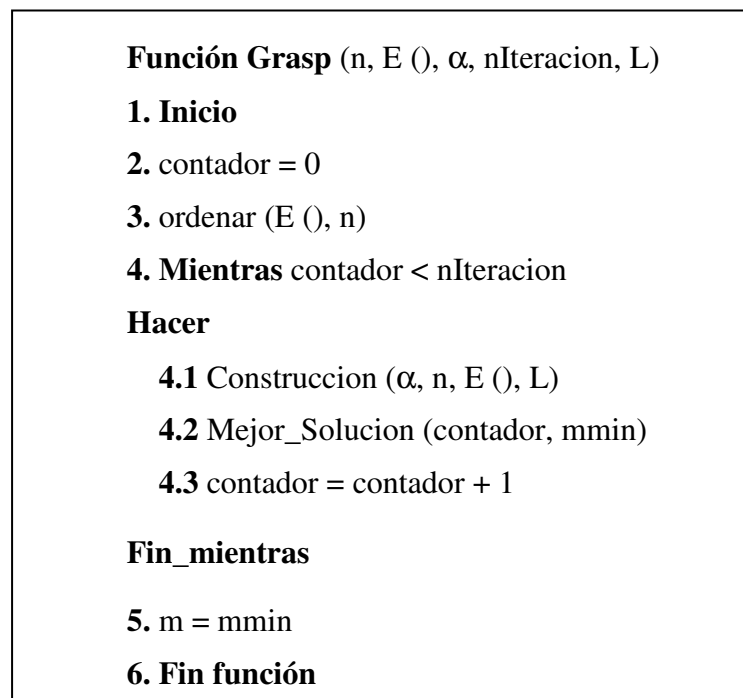


Figura 3.6. Algoritmo GRASP para el problema de cortes 1D [Mauricio+02]

Los parámetros de entrada que se observa en el algoritmo son:

- La constante  $n$ , almacena el número de requerimientos, es decir, el número de piezas.
- El arreglo  $E$ , almacena los tamaños requeridos para cubrir la demanda, es decir, el tamaño de cada pieza.
- La constante  $L$ , almacena la longitud estándar de las barras a utilizar.
- El parámetro de relajación  $\alpha$  es utilizado en la función *Construccion*.
- La constante  $nIteracion$ , almacena el número máximo de iteraciones permitidas, el cual es usado como criterio de parada.
- La variable  $m$ , almacena la mejor solución.
- La variable  $mmin$ , almacena la mejor solución obtenida en cada iteración

En las líneas 2 y 3 de la Figura 3.6 se inicializan las variables de control de la iteración y se ordena el conjunto de requerimientos  $E$  de forma descendente respectivamente.

En la línea 4 de la Figura 3.6 se muestra la condición de parada que el algoritmo utiliza para encontrar una solución óptima.

En las líneas 4.1, 4.2 y 4.3 de la Figura 3.6 se procede con la construcción de la LRC, para procesar las piezas a cortar de forma iterativa, donde se obtiene una solución y se procede a registrar una mejor solución si este fuera el caso, luego el *contador* suma una unidad por cada iteración GRASP.

Por último, en la línea 5 de la Figura 3.6 se almacena el resultado en  $m$ , que es la mejor solución encontrada hasta el momento.

### Complejidad del GRASP

En la implementación de [Ganoza+02] se obtiene la complejidad del GRASP *Construcción* de:  $O(n^2)$

#### 3.3.3.2 Algoritmos Genéticos

Los algoritmos genéticos existentes en la literatura para el problemas de cortes 1D son diversos, donde se muestra las distintas implementaciones y comparación de resultados de las mismas.

Podemos observar estos algoritmos en las siguientes publicaciones de [karelahti02], [Khalifa06] donde se detalla los pasos de un algoritmo genético [Torres03].

## **3.4 Aplicativos**

### **3.4.1 1DNest Versión 3.3**

**Descripción:**

Herramienta para optimizar el aprovechamiento de materiales cuando se requiere cortar diversos trozos más pequeños desde materiales que son provistos en longitudes estandarizadas, como es el caso de perfiles metálicos, extrusiones, vigas, cañerías, tubos, rollos de papel, etc.

**Empresa y/o Web site:**

<http://www.1d-solutions.com/>

**Bondades:**

Considera tolerancia de cortes por el espesor de la herramienta de corte.

Puede incluir hasta 150 largos diferentes de trozos a cortar.

Puede incluir hasta 50 distintos insumos (barras) para trozar.

Detalle de ángulos de corte en los trozos.

Holgura para considerar las irregularidades en extremos de las barras.

**Funcionalidad / Método:**

Utiliza un algoritmo de optimización basado en un método exacto, un nuevo algoritmo complementario permite resolver en tiempo real y en forma transparente los casos con grandes cantidades de datos que no son accesibles (por limitación de tiempo y/o memoria) al algoritmo exacto.

**Restricción:**

Puede ingresar hasta 1000 trozos diferentes a cortar.

Trabaja sobre plataformas: Windows 95, Windows NT, Windows 2000, Windows XP, Windows 98, Windows ME, Windows Vista.



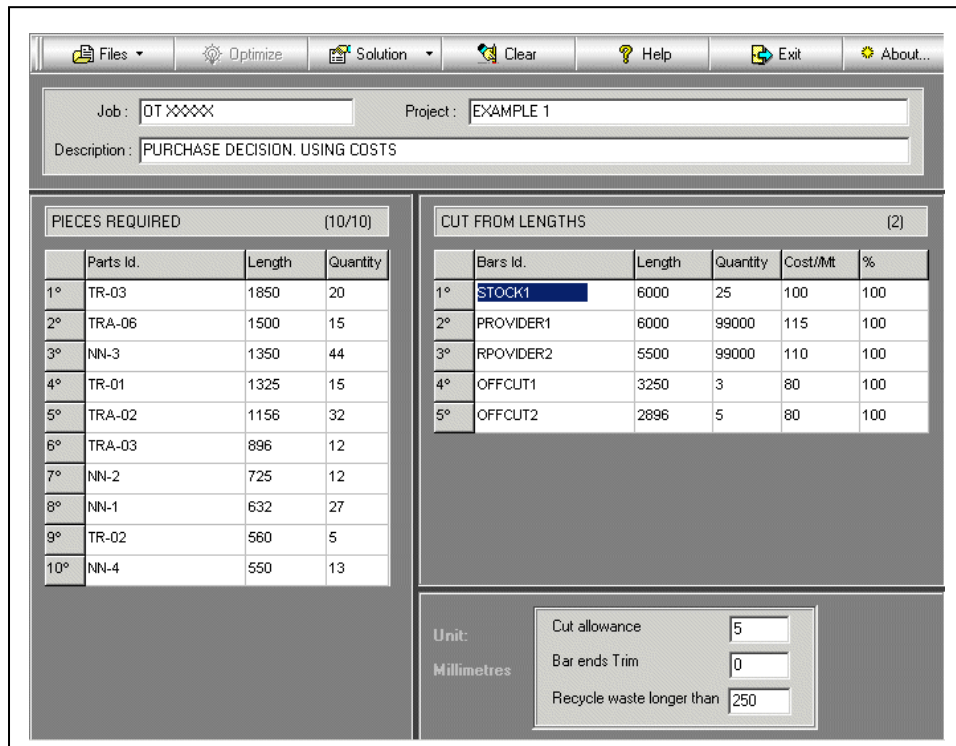


Figura 3.7. Aplicativo 1DNest Versión 3.3.

### 3.4.2 1D Stock Cutter

#### Descripción:

Software desarrollado para planear el diseño de cualquier corte de material lineal.

#### Empresa y/o Web Site:

Astrokettle Algorithms - <http://www.astrokettle.com/index.html>

#### Bondades:

El programa reduce el desperdicio del cerca de 10%.

Importación y exportación en archivos de texto.

Salida gráfica y los informes completos.

#### Funcionalidad / Método:

El método usado para su desarrollo se basa en Algoritmos Genéticos.

#### Restricción:

Funciona bajo los sistemas operativos: Windows 95, Windows 98, Windows ME, Windows NT 4.x, Windows 2000.

El máximo total de piezas a cortar es 5000.

El número total de piezas de diferente tamaño es 100.

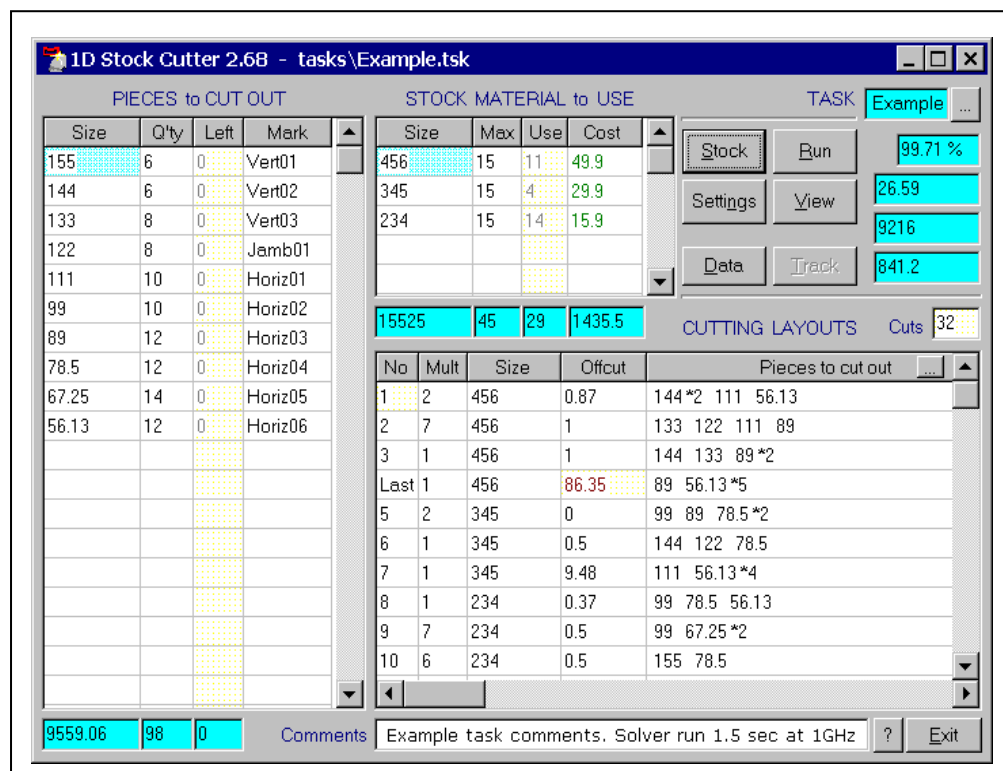


Figura 3.8. Aplicativo 1D Stock Cutter.

### 3.4.3 CutLogic 1D

#### Descripción:

Se ahorrará material, trabajo y tiempo cuando se necesite cortar cualquier material lineal tal como pipas, perfiles, barras, cables etc. El programa proporciona el corte común excelente, lo cual reduce al mínimo el coste total, así mismo, disminuye el desperdicio al mínimo, lo cual reduce al mínimo el número de las disposiciones del corte, más la optimización de los otros 4 valores, todos simultáneamente.

#### Empresa y/o Web Site:

TMachines - <http://www.tmachines.com/cutlogic-1d.htm>

#### Bondades:

Formato de longitud variable.

Importar, exporta resultados.

**Funcionalidad / Método:**

No describe.

**Restricción:**

Requerimientos mínimos del sistema: Procesador Pentium o compatible, 256 MB RAM, 30 MB libre de espacio de disco duro y Monitor con Resolución 800x600.

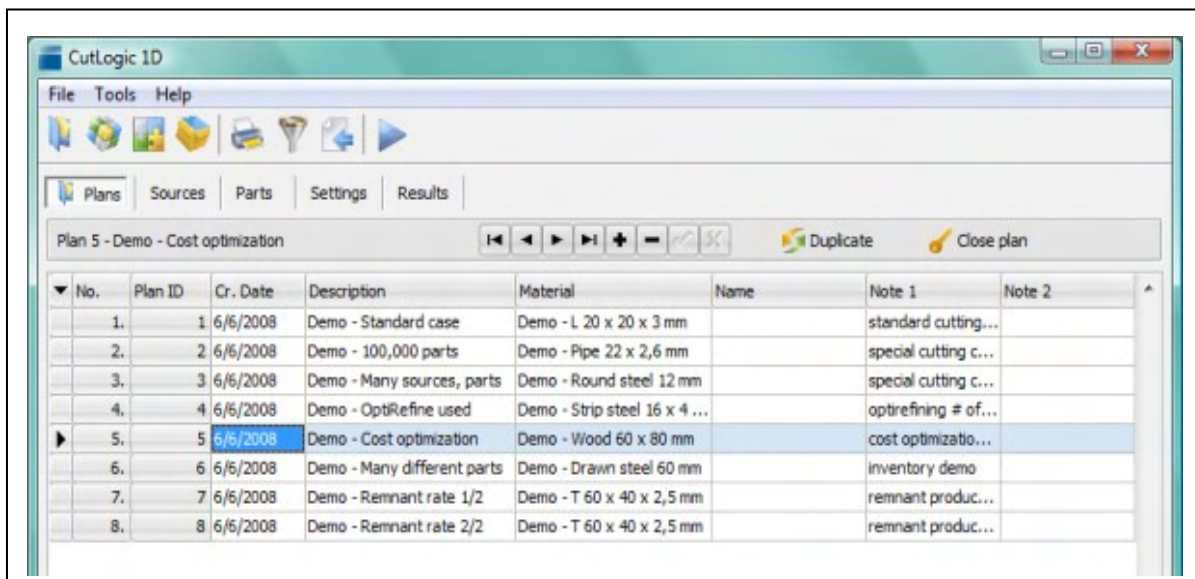


Figura 3.9. Aplicativo CutLogic 1D.

**3.4.4 Real Cut 1D 7.8.5.0**

**Descripción:**

Real Cut 1D es un programa de computadora usado para obtener óptimos diseños de corte de piezas unidimensionales con ángulos diferentes a 90 grados en las extremidades. Real Cut 1D puede ser usado para cortar piezas lineales como barras, tuberías, tubos, barras de acero, etc.

**Empresa y/o Web Site:**

<http://www.optimalprograms.com/>

**Bondades:**

Las piezas a cortar pueden tener ángulos diferentes de 90 grados en sus extremidades. Mucho material puede ser ahorrado al igualar 2 piezas que tengan extremidades afiladas.

Métrica de medida independiente.

Grosor ajustable de la hoja de cortar.

Nivel de optimización ajustable.

Impresión de los diseños obtenidos.

Impresión del REPOSITORIO y de la DEMANDA del cliente.

Estadísticas relacionadas con el corte actual.

Tiempo de ejecución muy rápido (las soluciones son obtenidas en segundos).

Los datos pueden ser exportados/importados desde Excel, Word, y archivos XML.

### Funcionalidad / Método:

No describe.

### Restricción:

No describe.

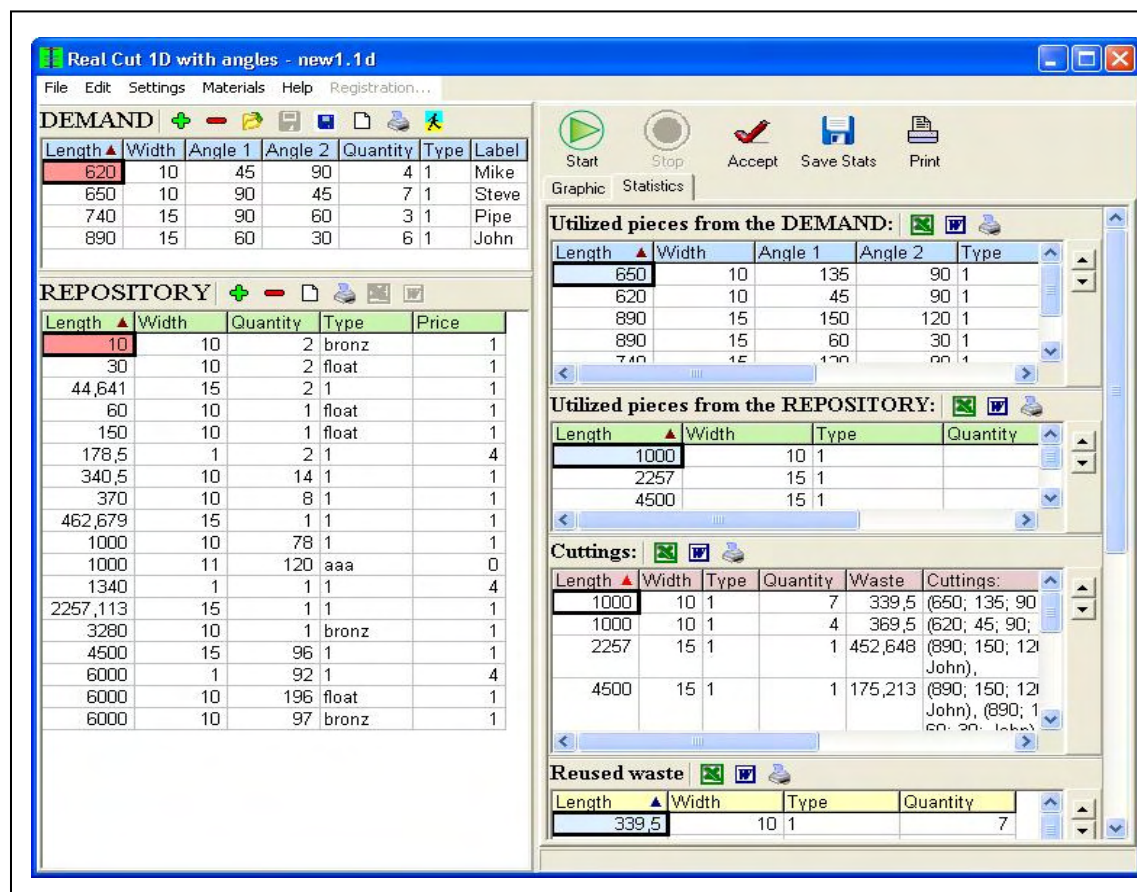


Figura 3.10. Aplicativo Real Cut 1D 7.8.5.0.

## Capítulo 4: Un Algoritmo GRASP Reactivo

En este capítulo se propone un algoritmo GRASP Reactivo para resolver el problema de cortes 1D.

El algoritmo propuesto se basa en el trabajo de Ganoza y Solano [Ganoza+02], quienes desarrollaron un GRASP básico con un parámetro de relajación; y se basa también en el trabajo de [Mauricio+02] donde se propone algoritmos de GRASP construcción con dos parámetros de relajación, cuya adaptación se realiza en la selección dinámica para el parámetro  $\alpha$  y  $\theta$  del algoritmo GRASP.

### 4.1 Evaluación y selección del método

El GRASP muestra resultados satisfactorios implementando el algoritmo de manera muy simple, descritos en [Feo+95].

En la tesis de [Vieira04] se observa que los resultados de las pruebas realizadas para el problema de cortes 1D dependen del parámetro  $\alpha$ , en el cual se menciona que, para mejores resultados, el parámetro  $\alpha$  debe ser seleccionado de manera aleatoria y conforme a la cantidad de ítems que adoptará el problema.

En [Prais+98], se implementa inicialmente la variante reactiva obteniendo mejores resultados.

Por otro lado, en la tesis de [Gómez+01] se muestran resultados satisfactorios donde se compara el algoritmo GRASP básico con uno Reactivo y con Path Relinking a favor del GRASP Reactivo para un problema en redes de telefonía móvil. En [Caballero08], al aplicar GRASP Reactivo para el diseño eficiente de territorios de atención comercial con requerimientos de asignación conjunta, se observa una calidad y diversas soluciones eliminando el problema de ajuste específico para escenarios con características diferentes.

En la literatura, tal como se puede apreciar, se han obtenido mejores resultados utilizando el GRASP Reactivo, cuya característica fundamental es la autorregulación del parámetro  $\alpha$  de acuerdo a la calidad de soluciones previamente obtenidas por cada una de ellas [Resende03], y que ha sido implementado satisfactoriamente en otras

variantes del problema de cortes [Delorme+03], [Alvarez+08], pero que para el caso de 1D aún no ha sido implementada.

Por lo tanto, se selecciona el GRASP Reactivo con el objetivo de resolver el problema de cortes 1D y mejorar los resultados obtenidos en [Ganoza+02].

## 4.2 Definición de variables y parámetros

Se define el siguiente listado de variables y parámetros para describir el problema de cortes 1D y el algoritmo GRASP reactivo con 2 parámetros de relajación:

- $L$ : Longitud estándar de la barra a utilizar.
- $n$ : Número total de cortes requeridos (cantidad de piezas).
- $E(i)$ : Tamaño del requerimiento “ $i$ ”.
- $B$ : Solución del problema mostrando las barras utilizadas.
- $R$ : Resíduo de cada barra utilizada.
- $i$ : Es el requerimiento “ $i$ ”, donde  $0 < i \leq n$ , entero.
- $\alpha$ : Es el parámetro utilizado en el procedimiento de Construcción del GRASP reactivo como parámetro de relajación.
- $\theta$ : Es el parámetro utilizado en el procedimiento de Construcción del GRASP reactivo para la selección aleatoria de la barra a cortar.
- $Z^*$ : Es la mejor solución obtenida hasta el momento con los parámetros  $\alpha$  y  $\theta$ .
- $D(j)$ : Son los valores discretos predefinidos de  $\alpha$  o  $\theta$ .
- $A(j)$ : Es el valor promedio de soluciones obtenidas utilizando  $\alpha$  o  $\theta$ .
- $P(j)$ : Es la probabilidad del parámetro  $\alpha$  o  $\theta$ .
- $q(j)$ : Es el valor que contiene la relación  $Z^*/A_j$  para  $j = 1..h$ .
- $\phi$ : Parámetro para atenuar la magnitud de los valores de  $q(j)$ .
- $nIteracion$ : Es una constante que guarda la máxima cantidad de iteraciones que realizará el algoritmo, utilizada como criterio de parada.
- $periodo\_act$ : Es una constante que permite actualizar periódicamente las probabilidades de  $P(j)$ .

### 4.3 Estructura de Datos

La estructura de datos servirá para desarrollar e implementar el GRASP, que se describe a continuación, con el fin de almacenar la información necesaria para el funcionamiento del programa, tal como se muestra en [Ganoza+02], y también para almacenar las estructuras definidas para el algoritmo GRASP Reactivo.

- *Arreglo de cortes.* Este arreglo se muestra de la forma  $E : arreglo[1..n]$ . Se almacenará los tamaños que se requieren cortar para satisfacer la demanda.
- *Arreglo de Barras.* Se muestra la matriz para las barras  $B : matriz[1..m][1..c]$ . Esta matriz contendrá en cada fila los diferentes cortes que se han realizado en la barra de tamaño  $L$ , donde  $m$  el número de barras usadas para satisfacer el requerimiento de la demanda y  $c$  es la constante para indicar el número máximo de cortes realizado por barras.
- *Arreglo de residuos.* Se muestra el vector  $r : arreglo[1..m]$ . El arreglo  $r$  contendrá los residuos de los cortes que se han realizado de las barras del arreglo  $B$  donde  $m$  es el número de barras usadas para satisfacer el requerimiento de la demanda.
- *Arreglo de parámetros  $\alpha_j$ .* Se describe el vector  $D : arreglo[1..h]$ . Este almacenará los distintos valores predefinidos para el parámetro de relajación  $\alpha$  donde  $h$  es la cantidad total de valores  $\alpha$  que se seleccionarán.
- *Arreglo de probabilidades.* Se muestra el arreglo de probabilidades  $P : arreglo[1..h]$ . El vector  $P$  almacenará las probabilidades relacionadas al parámetro de relajación  $\alpha$  en el arreglo  $D$ .
- *Arreglo de valores promedios de soluciones obtenidas con  $\alpha_i$ .* Se describe el arreglo  $A : arreglo[1..h]$ . Contendrá los valores promedios de las soluciones obtenidas hasta el momento, tras utilizar un valor  $\alpha$  del arreglo  $D$ .
- *Arreglo  $q$ .* Describe el arreglo  $q : arreglo[1..h]$ . El vector  $q$  contendrá la relación de dividir la solución óptima encontrada hasta el momento entre el valor promedio de las soluciones encontradas al utilizar el parámetro  $\alpha$  obtenida del arreglo  $A$ .

## 4.4 GRASP Reactivo

### 4.4.1 Algoritmo Principal

Inicialmente se ordenan los requerimientos en forma decreciente, luego entra en una iteración, seleccionando el parámetro  $\alpha$  y  $\theta$ . Luego se procede a la fase de construcción para obtener una solución inicial donde se usan los parámetros seleccionados, luego se mejora la solución, es decir, se trata de reducir la cantidad de barras utilizadas. Finalmente, si se logra mejorar, ésta será registrada como mejor solución actual.

Se detalla el algoritmo GRASP reactivo que se adaptará:

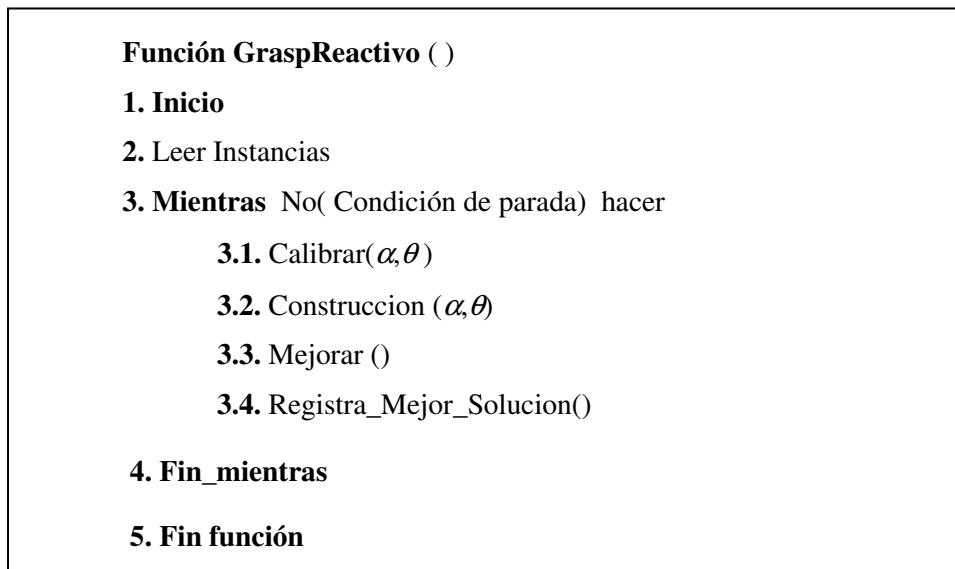


Figura 4.1. Algoritmo GRASP Reactivo para el problema de cortes 1D.

### Descripción del algoritmo

En la línea 2 se leen las piezas a cortar.

En la línea 3 se verifica si cumple la condición de parada.

En 3.1 se calibra los parámetros  $\alpha$  y  $\theta$ .

En 3.2 se construye la una solución inicial.

En 3.3 se mejora la solución obtenida en construcción.

En 3.4 se registra la mejor solución.



#### 4.4.2 Algoritmo Construcción

La funcionalidad principal del GRASP reactivo se desarrolla en este procedimiento de construcción ya que se encuentra una solución para el problema, donde se aplica como criterio de selección tanto el algoritmo goloso FFD como el BFD, generando una lista de soluciones candidatas y se seleccionará aleatoriamente una pieza de esta lista que no será necesariamente la pieza de mayor tamaño [Ganoza+02], generando otra lista de barras candidatas ya cortadas donde se seleccionará una de ellas.

Se describe el algoritmo *Construcción*:

**Construccion ( $\alpha$ ,  $n$ ,  $E$ ,  $L$ ,  $\theta$ )**

- 1. Inicio**
- 2.**  $m = 0$
- 3.**  $r(1) = L$
- 4. Mientras**  $n > 0$  **hacer**
  - 4.1**  $num\_lrc = 0$ ;  $num\_lrc\_B = 0$ ; ordenar ( $r()$ ,  $B()$ ,  $m$ )
  - 4.2**  $\beta = E(0)$ ;  $d = E(n - 1)$ ;  $R_{min} = r(m - 1)$
  - 4.3**  $llenar\_LRC(E(), n, \alpha, \beta, d, lrc(), num\_lrc)$
  - 4.4**  $llenar\_LRC\_B(r(), m, \theta, R_{min}, lrc\_B(), num\_lrc\_B)$
  - 4.5**  $num\_lrc = \text{Redondeo}(num\_lrc * Rnd)$
  - 4.6**  $num\_lrc\_B = \text{Redondeo}(num\_lrc\_B * Rnd)$
  - 4.7**  $resultado = lrc(num\_lrc)$
  - 4.8**  $k = lrc\_B(num\_lrc\_B)$
  - 4.9**  $eliminar\_resultado(E(), n, num\_lrc)$
  - 4.10. Si**  $k = m + 1$  **Entonces**
    - 4.10.1.**  $m = m + 1$
    - 4.10.2.**  $b(k)(1) = resultado$
    - 4.10.3.**  $r(k) = L - resultado$
    - 4.10.** **Sino**  $j = 1$
    - 4.10.5.** **Mientras**  $b(k, j) > 0$ 
      - $j = j + 1$
      - fin mientras**
    - $b(k, j) = resultado$
    - 4.10.6.**  $r(k) = r(k) - resultado$
  - 4.11. Fin Si**
- 5. Fin Mientras**
- 6. Fin Construccion**

Figura 4.2. Algoritmo GRASP Construcción BFD para el problema de cortes 1D.

### Descripción del algoritmo Construccion

El algoritmo Construccion seleccionará una barra de LRC\_B generado por la función *llenar\_LRC\_B* y una pieza a cortar de LRC generado por la función *llenar\_LRC*, de

forma aleatoria utilizando los parámetros  $\alpha$  y  $\theta$ , respectivamente. Luego se eliminará la pieza seleccionada del listado de  $E$  y se cortará la pieza en la barra seleccionada.

La función *llenar\_LRC* que se muestra en la línea 4.3 de la Figura 4.2, generará la lista restringida de candidatos de las posibles piezas a procesar, y los guardará en el vector *lrc* de tamaño *num\_lrc*.

Se consideran los siguientes criterios para su construcción:

- Encontrar un valor  $d$  que representa el menor tamaño del requerimiento o pieza no atendida.
- Encontrar un valor  $\beta$  que equivale al mayor tamaño de los requerimientos.
- Encontrar  $\alpha$ , parámetro de relajación asociado a la selección de la pieza a cortar, donde el valor puede ser entre 0 y 1.

Se presenta el algoritmo *llenar\_LRC* a implementar. Este algoritmo agrupa los posibles requerimientos o piezas que serán seleccionados con el criterio goloso – aleatorio según el valor de  $\alpha$ :

**Función llenar\_LRC (E, n,  $\alpha$ ,  $\beta$ , d, lrc, num\_lrc)**

1. Entero ee
2. Para ee = 0 hasta n – 1 hacer
  - 2.1. Si  $E(ee) \geq \beta - \alpha * (\beta - d)$  entonces
    - 2.1.1.  $lrc(num\_lrc) = E(ee)$
    - 2.1.2.  $num\_lrc = num\_lrc + 1$
  - 2.2. Sino
    - 2.2.1. ee = n - 1
  - 2.3. Fin Si
3. Fin Para
4. Fin llenar\_LRC

**Figura 4.3. Función que genera la LRC.**

En las líneas 4.5 y 4.7 de la Figura 4.2 del algoritmo construcción se selecciona aleatoriamente un candidato y se guarda en la variable *resultado*.

La función *llenar\_LRC\_B*, que se muestra en la línea 4.4 de la Figura 4.2, generará la lista restringida de candidatos de las posibles barras que se seleccionará para el corte de las piezas y los guardará en el vector *lrc\_B* de tamaño *num\_lrc\_B*.

Se consideran los siguientes criterios para su construcción:

- Ordenar de forma decreciente, de mayor a menor los residuos de cada barra.
- Encontrar un valor  $R_{min}$  que representa el menor desperdicio de las barras cortadas hasta el momento
- Encontrar  $\theta$ , parámetro de relajación asociado a la selección de la barra a cortar, donde el valor puede ser entre 0 y 1.

Se presenta el algoritmo *llenar\_LRC\_B* a implementar. Este algoritmo agrupa las posibles barras que serán seleccionados con el criterio goloso - aleatorio según el valor de  $\theta$ :

**Función llenar\_LRC\_B (r, m,  $R_{min}$ ,  $\theta$ , lrc\_B, num\_lrc\_B)**

**1. Entero ee**

**2. Para ee = 0 hasta m – 1 hacer**

**2.1. Si  $r(ee) \leq R_{min} + \theta * (L - R_{min})$  y  $r(ee) \geq R_{min}$  entonces**

**2.1.1.  $lrc\_B(num\_lrc\_B) = r(ee)$**

**2.1.2.  $num\_lrc\_B = num\_lrc\_B + 1$**

**2.2. Sino**

**2.2.1.  $ee = m - 1$**

**2.3. Fin Si**

**3. Fin Para**

**Figura 4.4. Función que genera la LRC\_B.**

En la línea 4.8 de la Figura 4.2 se obtiene la barra a cortar seleccionada aleatoriamente. También se detallará, en 4.9 de la Figura 4.2, la función *eliminar\_resultado*, que eliminará el valor de la variable *resultado* en la lista de piezas *E*, hallada en la línea 4.3 de la función *llenar\_LRC*:

**Función Eliminar\_resultado (E, n, num\_lrc)**

**1. Para i = num\_lrc hasta n-1**

**1.1.  $E(i) = E(i+1)$**

**2. Fin Para**

**3.  $n = n-1$**

**4. Fin Eliminar\_resultado**

**Figura 4.5. Función eliminar resultado.**

Luego, en las líneas del 4.10 al 4.11 de la Figura 4.2, se realiza el corte de la pieza en la barra seleccionada, actualizando las barras y sus respectivos residuos conforme se hayan utilizado nuevas barras o barras ya usadas.

### 4.4.3 Algoritmo Mejora

Se encarga de mejorar la solución encontrada hasta el momento, esto es, que la solución pueda ser mejorada utilizando una menor cantidad de barras usadas.

Como parámetro de entrada,  $mmin$  almacenará la cantidad de barras utilizadas en la mejor solución, y  $L$  la longitud máxima de cada barra.

El algoritmo se basa bajo el criterio descrito en [Mauricio04] en el que se consideran tres pasos:

1. Seleccionar la barra que presenta mayor desperdicio denotándolo como  $Br1$ .
2. Seleccionar la barra que presenta mayor desperdicio sin considerar  $Br1$ , denotándolo como  $Br2$ .
3. Intercambiar piezas entre la barra  $Br2$  y las demás piezas de las otras barras restantes, de manera que lo que resta en  $Br2$  o en las otras barras restantes, nos permita encajar una pieza de la barra  $Br1$ .

Estos pasos se repetirán hasta que se puedan encajar todas las piezas de la barra  $Br1$ , esto es, hasta haber disminuido la barra utilizada, en consecuencia se puede mejorar la solución.

La figura siguiente representa al algoritmo Mejora:

```
Mejora ( mmin, L)  
1. Inicio  
2. flag_mejora = falso;  
3. localIter = 0;  
4. Mientras flag_mejora=falso y localIter<=3 Hacer  
    4.1. k=1 ; j = 1;  
    4.2. Br1 = mayor_resto(b(), r>(), k, j )  
    4.3. Br2 = mayor_resto( b(), r(),k,j)  
    4.4. Intercambiar(Br1(), Br2(),b(), k, L, mmin)  
    4.5. Si Br1 no existe  
        4.5.1. flag_mejora = verdadero  
        4.5.2. Sino  
            4.5.2.1. localIter = localIter+1  
        4.5.3. Fin Si  
5. Fin Mientras  
6. Fin GRASP_Mejora
```

**Figura 4.6. Función Mejora.**

#### **4.4.4 Algoritmo Registra\_Mejor\_Solucion**

Esta función registra y actualiza la mejor solución encontrada hasta el momento, esto es, que la solución posea una menor cantidad de barras usadas. La Figura 4.7 representa a la función *Registra\_Mejor\_Solucion*:

**Registra\_Mejor\_Solución (contador, mmin)**

**1. Inicio**

**2. Si  $m < mmin$  o contador = 0 Hacer**

**2.1. Para  $i = 0$  hasta  $m$**

**2.1.1. Mientras  $b(i)(j) > 0$  hacer**

**2.1.1.1.  $b\_min(i)(j) = b(i)(j)$**

**2.1.1.2.  $r\_min(i) = r(i)$**

**2.1.1.3.  $j=j+1$**

**2.1.2. Fin mientras**

**2.2. Fin para**

**3.  $mmin = m$**

**4. Fin si**

**5. Fin GRASP\_Registra\_Mejor\_Solucion**

Figura 4.7. Función Mejor Solución.

**Descripción del algoritmo Registra\_Mejor\_Solucion**

Como entrada se tiene al parámetro *contador* que proporciona información que indica si existe alguna solución para realizar una comparación.

Como segundo parámetro de entrada *mmin* almacenará la cantidad de barras utilizadas en la mejor solución.

El registro consiste en comparar la solución actual encontrada *m* con la mejor solución actual *mmin*, de tal forma que si *m* es menor se procederá a actualizar la solución por la mejor encontrada.

**4.4.5 Descripción del algoritmo Calibrar**

Este algoritmo se encargará de actualizar bajo una condición, el periodo de iteraciones que se contemple, para las probabilidades del arreglo *P* para cada parámetro  $\alpha$  ó  $\theta$  correspondiente. Se detalla el algoritmo:

**Funcion Calibrar( $\alpha, \theta$ )**

- 1.Inicio**
- 2. Si ( nIteracion mod periodo\_act ) = 0 Hacer**
  - 2.1.**  $A(j) = \text{Sum}(j) / \text{Cant}(j)$
  - 2.2.**  $q(j) = (\text{mmin} / A(j) )^\varphi$
  - 2.3.**  $P(j) = q(j) / \sum_{t=1}^h q(t)$
- 3. Fin Si**
- 4.Fin Calibrar**

**Figura 4.8. Función Calibrar parámetros de relajación.**



## **Capítulo 5: Sistema de optimización de corte 1D**

En este capítulo se describe el sistema implementado, se contempla la implementación del GRASP básico y GRASP Reactivo, se detalla los requerimientos mínimos de hardware y software para su funcionamiento, la estructura del sistema y una descripción breve de cada uno de los módulos que comprende.

### **5.1 Requerimientos mínimos de hardware y software**

#### **5.1.1 Configuración del hardware**

La configuración hardware mínimo que se requiere para que el sistema trabaje adecuadamente es la siguiente:

- Procesador: Pentium IV o superior.
- Velocidad: 1.7 GHz o superior.
- Memoria: 256MB de memoria RAM.
- Lectora CD.
- Mouse y teclado.

#### **5.1.2 Configuración del software**

Configuración mínima del software utilizado para la ejecución del sistema:

- Jdk 1.6.0
- Sistema Operativo Microsoft Windows XP, Profesional 2000, 2000 Server, Vista
- MS Office 2000, XP, 2003

### **5.2 Descripción del Sistema**

El sistema para resolver el Problema de Cortes 1D tendrá el nombre de GraspCorte, se implementa la meta heurística GRASP Reactiva, con el procedimiento de Construcción (mediante el criterio FFD y BFD) y el algoritmo de Mejora, descrita en el capítulo 4, así mismo se implementa la meta heurística GRASP básica.

Por otro lado, los valores de los parámetros que utiliza el sistema se almacenarán en archivos:

- **barra.txt** : almacena la longitud de la barra.
- **grasp.txt**: almacena los parámetros que requiere la meta heurística GRASP: Criterio de selección FFD o BFD, número de iteraciones y valor del parámetro de relajación  $\alpha$  y  $\theta$ .
- **graspreactivo.txt**: almacena los parámetros que requiere la meta heurística GRASP Reactivo: Criterio de selección FFD o BFD, número de iteraciones, conjunto de parámetros de relajación, periodo de actualización de las probabilidades de  $\alpha$  y  $\theta$ .

El software ha sido desarrollado mediante las siguientes herramientas:

- IDE NetBeans 6.5
- JasperReport 3.5, para la generación de reportes
- Ireport 3.0, para el diseño de reportes
- JDK 1.6.0
- Office 2003, MS Excel para la lectura de datos

### 5.2.1 Estructura del Sistema

El sistema está compuesto por los siguientes módulos:

- Módulo de configuración de parámetros
- Módulo de carga de piezas
- Módulo de ejecución de GRASP
- Módulo de reportes

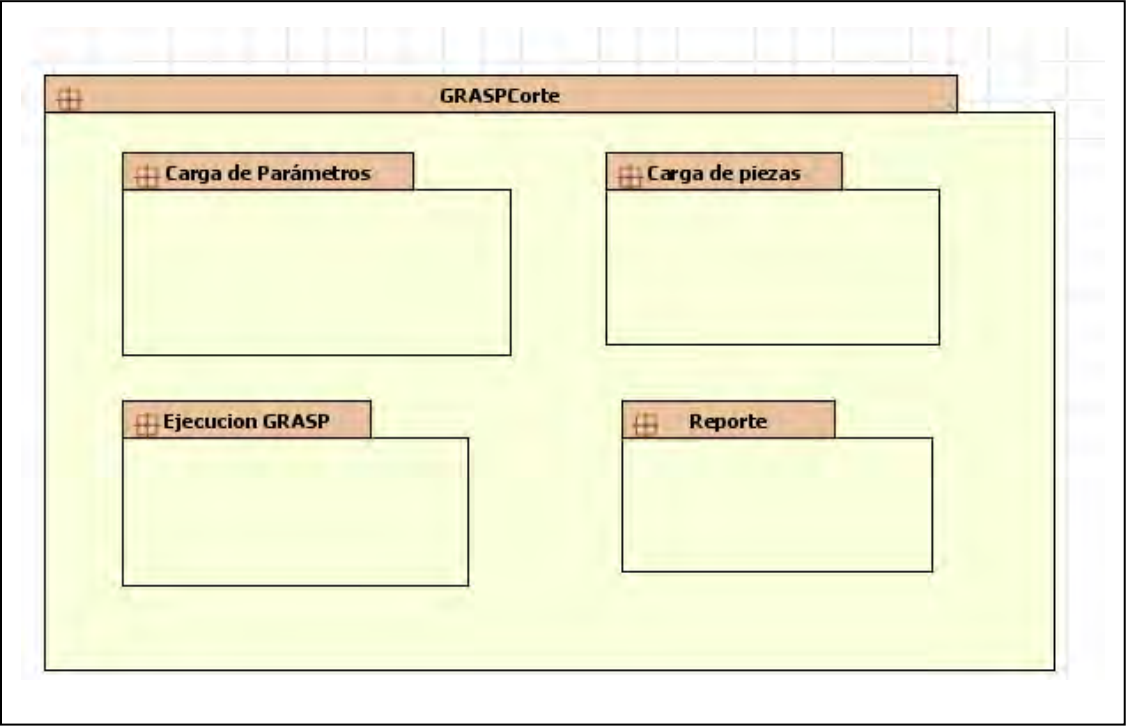


Figura 5.1. Módulos del sistema GraspCorte.

La pantalla principal del sistema GraspCorte se presenta a continuación

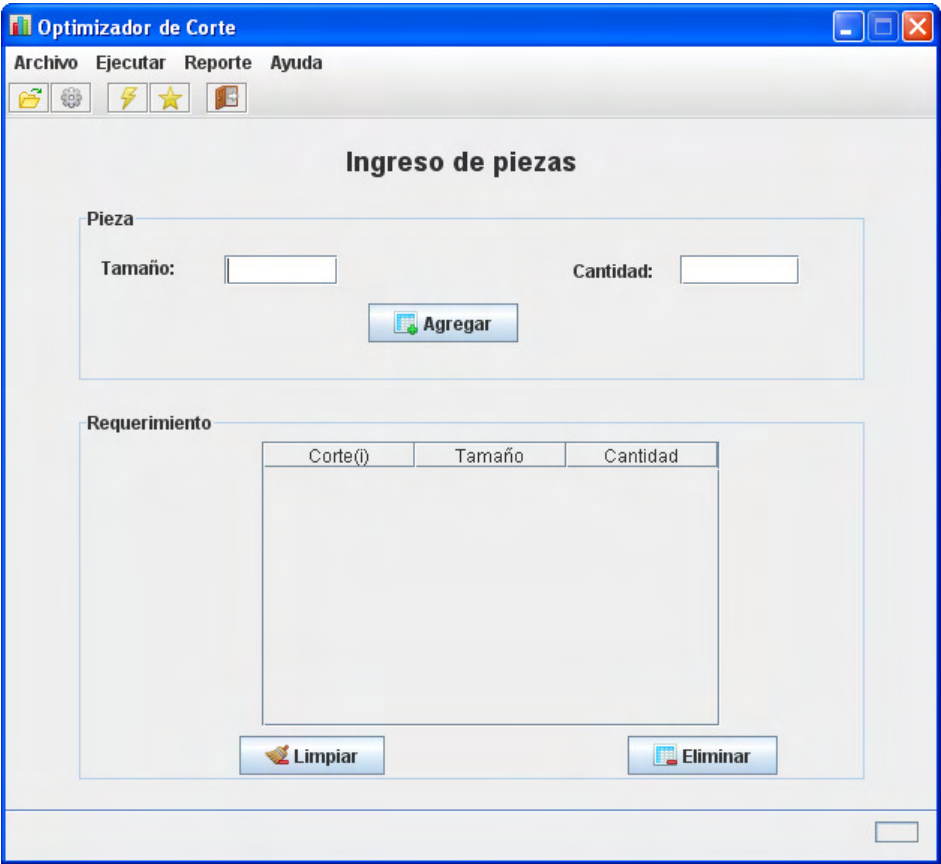


Figura 5.2. Ventana principal del sistema GraspCorte.

## 5.2.2 Módulos del Sistema

### 5.2.2.1 Módulo de configuración de parámetros

Permite el ingreso de los parámetros para la longitud de la barra estándar, el algoritmo GRASP básico y para el algoritmo GRASP Reactivo.



Figura 5.3. Ventana de configuración de la barra estándar del sistema GraspCorte.

En la Figura 5.3 se ingresa la longitud de la barra estándar.



Figura 5.4. Ventana de configuración del Grasp básico del sistema GraspCorte.

En la Figura 5.4 se ingresan los valores para el número de iteraciones, el valor de  $\alpha$  y el criterio del GRASP construcción (FFD o BFD).

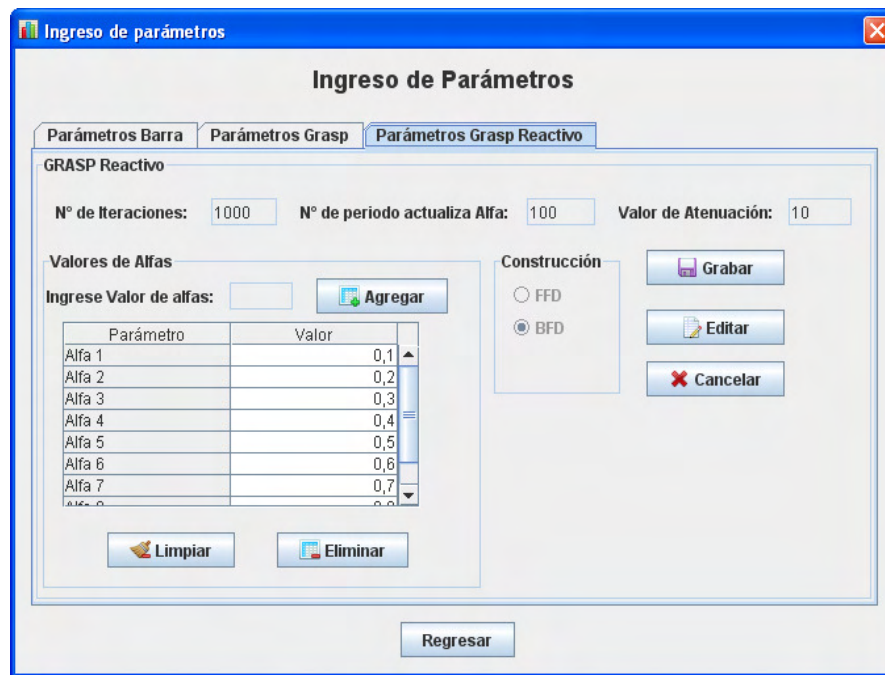


Figura 5.5. Ventana de configuración del Grasp Reactivo del sistema GraspCorte.

En la Figura 5.5 se ingresan los valores para el número de iteraciones GRASP, el número de iteraciones para el periodo de actualización de las probabilidades de  $\alpha$ , los  $\alpha$  que se considerarán, el criterio del GRASP construcción (FFD o BFD). Los valores para  $\theta$  serán los mismos valores que utilizará  $\alpha$

### 5.2.2.2 Módulo de Carga de piezas.

Permite al usuario ingresar al sistema el requerimiento de piezas de forma manual o ingresar cada pieza de forma automática, seleccionando un archivo Excel cuyo formato debe mostrar una columna del tamaño de cada pieza y otra columna con la demanda de cada pieza.

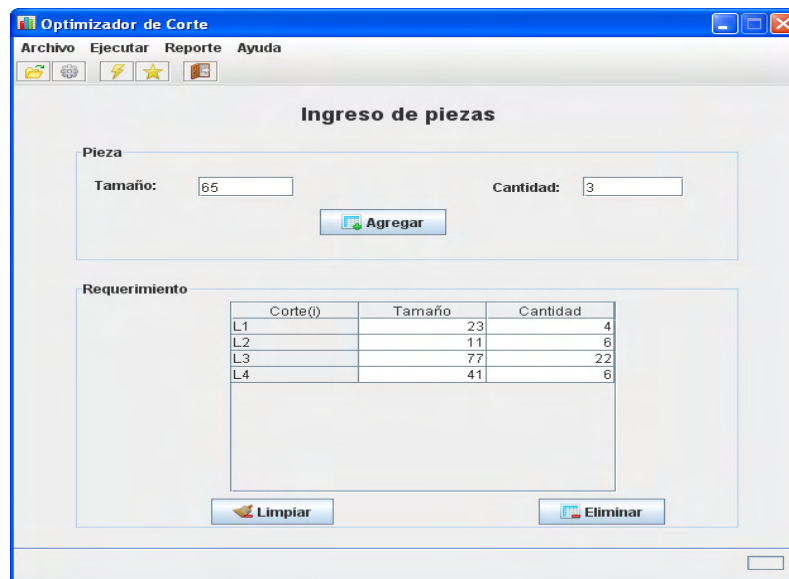


Figura 5.6. Ventana de carga manual de piezas del sistema GraspCorte.

En la Figura 5.6 se muestra el ingreso del tamaño de la pieza y la cantidad, presionando el botón *Agregar*.

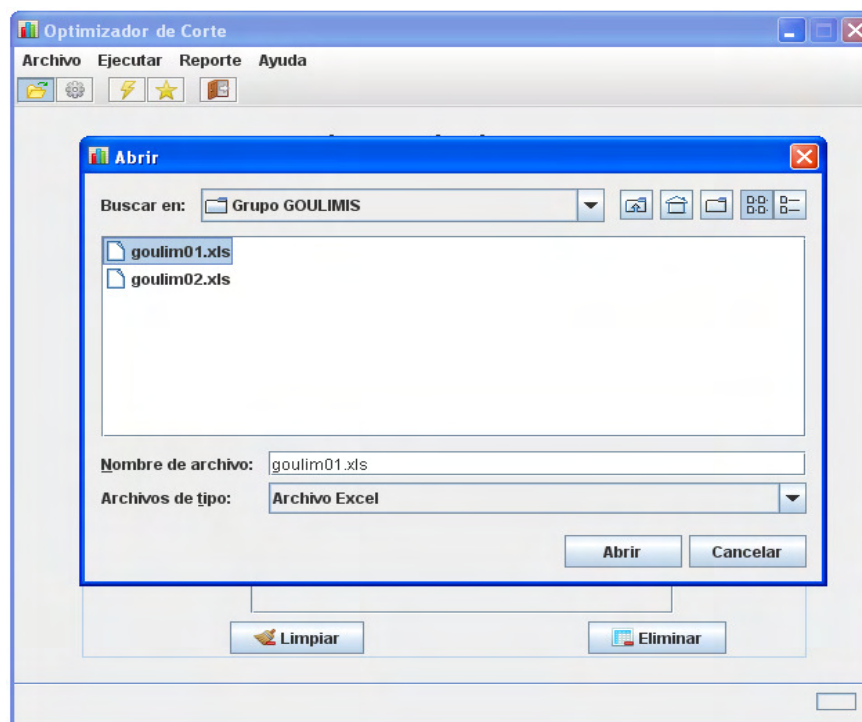


Figura 5.7. Ventana de carga automática de piezas del sistema GraspCorte.

En la Figura 5.7 se muestra la selección del archivo Excel que permitirá al sistema leer automáticamente las piezas.

### 5.2.2.3 Módulo de Ejecución del GRASP

El sistema permite la ejecución del GRASP básico mediante el botón , y del GRASP Reactivo mediante el botón , previo ingreso de piezas.

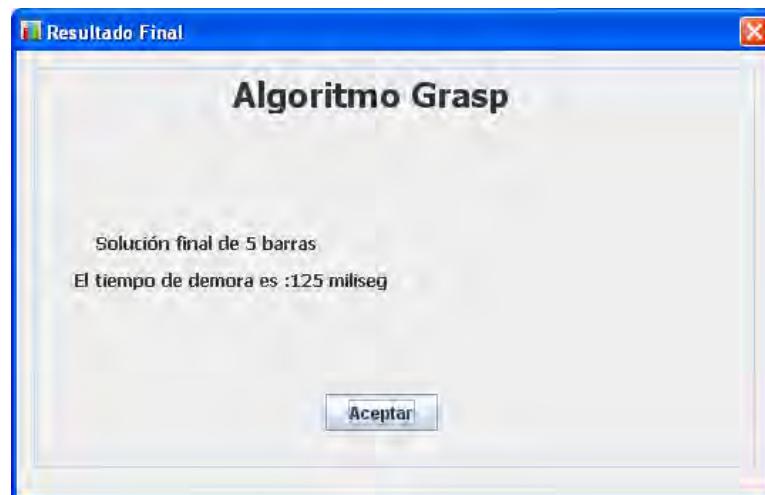


Figura 5.8. Resultado final del GRASP básico del sistema GraspCorte.

En la Figura 5.8 se muestra el resultado final al ejecutar el GRASP básico, indicando la cantidad de barras utilizadas y el tiempo que tomó la ejecución.

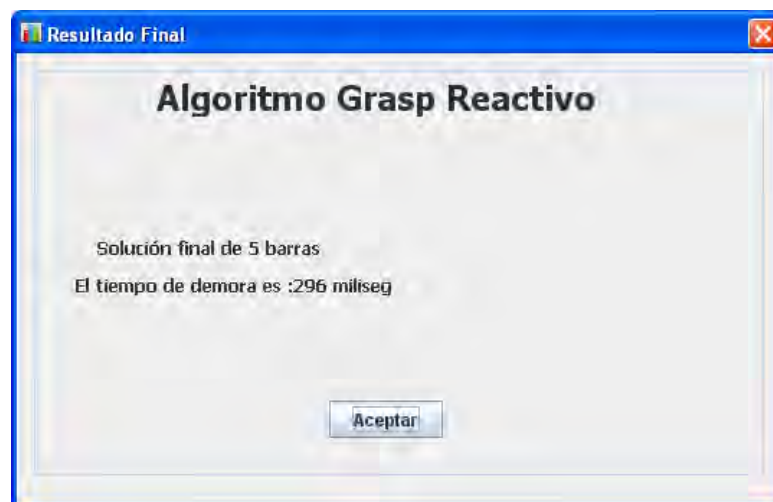
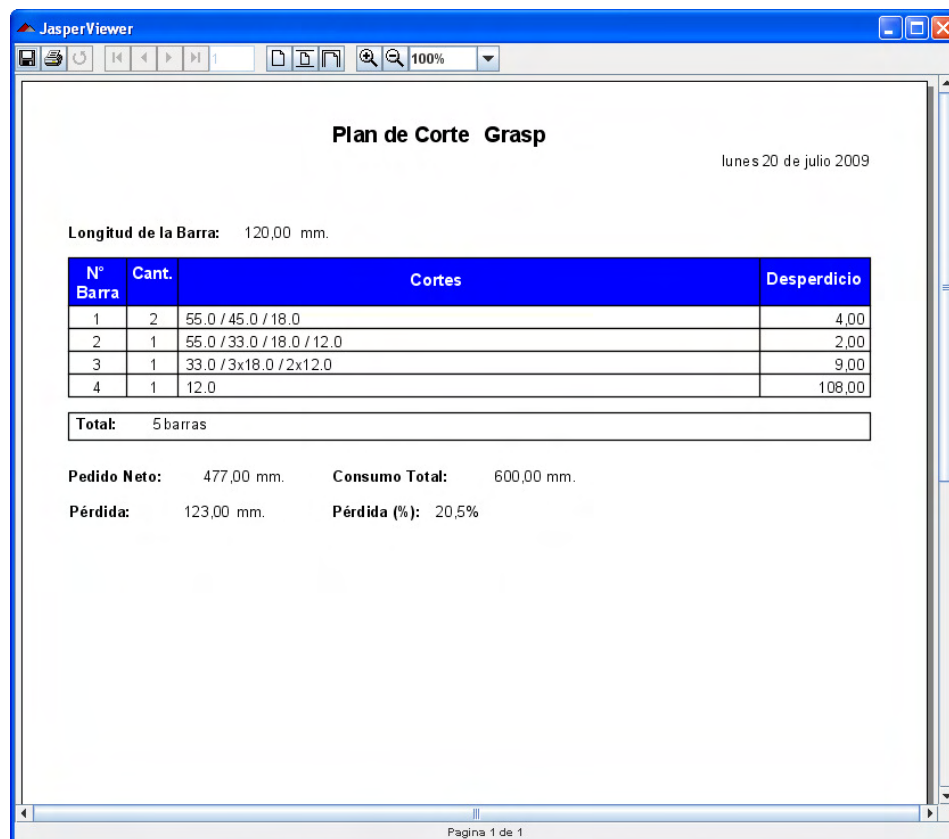


Figura 5.9. Resultado final del GRASP Reactivo del sistema GraspCorte.

En la Figura 5.9 se muestra el resultado final al ejecutar el GRASP Reactivo, donde se indica la cantidad de barras usadas y el tiempo que tomó la ejecución.

#### 5.2.2.4 Módulo de Reportes.

Permite al usuario visualizar el plan de corte en detalle, con los resultados obtenidos tras la ejecución del GRASP básico y GRASP Reactivo. Además se podrá imprimir o guardar para su posterior visualización.



**Figura 5.10. Reporte del GRASP básico del sistema GraspCorte.**

En la Figura 5.10 se detalla el reporte generado para el GRASP, en la primera columna se muestra el número de barra, en la segunda columna se indica la cantidad de barras utilizada, en la tercera columna se detalla los cortes realizados a la barra mostrando la cantidad y tamaño de la pieza atendida y en la última columna se muestra el desperdicio generado. También se indica la longitud total utilizada (pedido neto), consumo total, pérdida total y el porcentaje de pérdida.



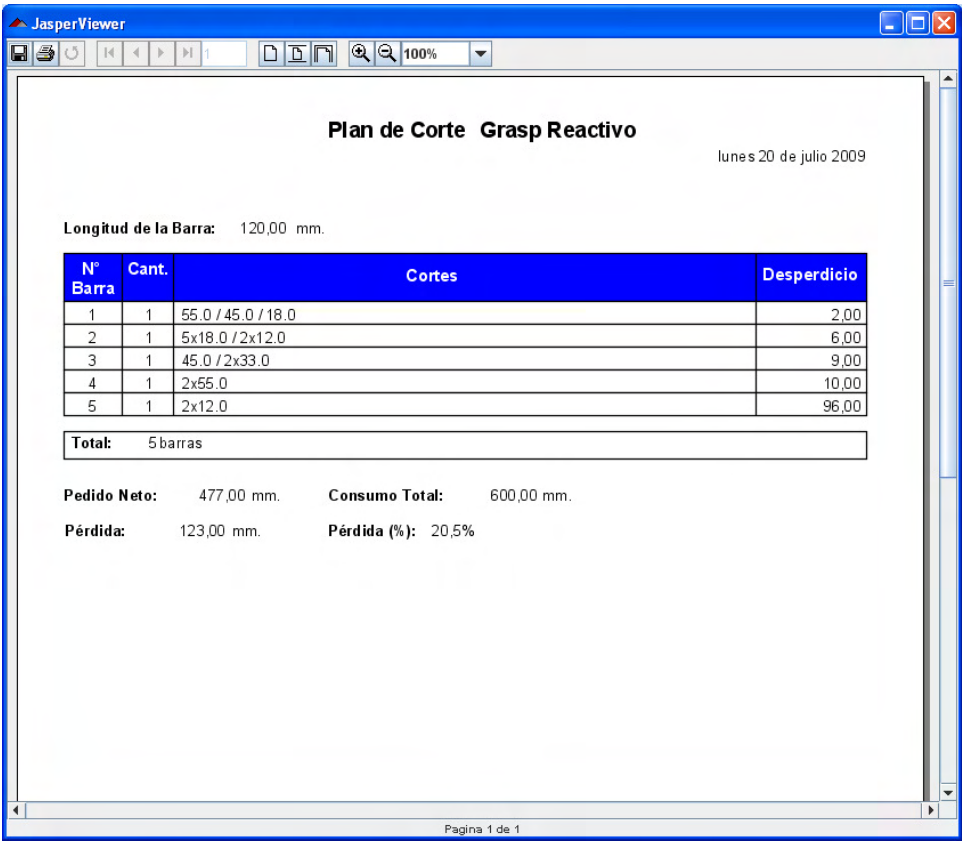


Figura 5.11. Reporte del GRASP Reactivo del sistema GraspCorte.

En la Figura 5.11 de describe de manera análoga el reporte de la Figura 5.10.

## Capítulo 6: Experimentos Numéricos

### 6.1 Hardware y software

Se detalla el hardware y software utilizado para el desarrollo del sistema de optimización de cortes 1D, GraspCorte.

#### 6.1.1 Hardware.

- Procesador: Intel® Core™ 2 Duo E4600
- Velocidad: 2.40 GHz
- Memoria: 3GB de memoria RAM
- Sistema de archivos: 32 bits

#### 6.1.2 Software.

- Sistema Operativo Microsoft Windows XP service Pack 2
- MS Office 2003 service Pack 3
- IDE NetBeans 6.5
- JasperReport 3.5, para la generación de reportes
- IReport 3.0.0, para el diseño del reporte
- JDK 1.6.0
- Office 2003, MS Excel para la lectura de datos

### 6.2 Instancias de prueba

Las instancias de prueba que se usaron para la demostración y el análisis de los resultados se obtuvieron de [Nacif97], [Stadtler90], [Mota89], [Degraeve+97], [Goulimis90], [Hinterding94] y [Aloise+91].

Se utilizaron como condición de parada para el algoritmo GRASP Reactivo la cantidad de 2000 iteraciones y como condición de optimalidad, que el desperdicio total sea menor a la longitud fija de una barra.

Para el GRASP básico se utiliza  $\alpha = 0.5$  y la condición de parada de 2000 iteraciones, aplicados en [Ganoza+02].

Se utilizan las siguientes abreviaturas para denominar los criterios del GRASP Construcción: FFD, BFD, FFDR (criterio FFD Reactivo), BFDR (Criterio BFD Reactivo).

Estas instancias se detallan en el anexo A y se agrupan de la siguiente forma:

Grupos	Autor	N° de Instancias
Grupo1	Nacif Rocha	10
Grupo2	Nacif Rocha	6
Grupo3	Nacif Rocha	2
Grupo4	Nacif Rocha	35
Grupo5	Stadtler	22
Grupo6	Pauta	2
Grupo7	Degraeve	2
Grupo8	Goulimis	2
Grupo9	Hinterding & Khan	4
Grupo10	Aloise & Maculan	15
Total:		100

**Tabla 6.1. Lista de Grupos de instancias de prueba.**

### 6.3 Resultados Numéricos

CODIGO	OPTIMO	GRASP CONSTRUCCION				GRASP CONSTRUCCION + MEJORIA			
		FFD	BFD	FFDR	BFDR	FFD	BFD	FFDR	BFDR
20-347	207	214	214	214	214	214	214	213	214
20-348	55	56	55	56	56	55	55	56	56
20-349	180	185	185	184	184	185	185	186	185
CM-063	142	146	146	146	146	146	146	146	146
CM-096	39	46	46	46	46	46	46	46	46
ME-001	78	81	80	80	80	80	80	80	80
ME-011	29	32	31	31	31	31	31	31	32
ME-028	186	191	190	190	190	190	190	190	190
ME-029	207	211	211	211	211	211	211	212	211
ME-030	292	299	298	298	298	298	298	298	298
<b>Eficiencia Promedio (%)</b>		95,00	95,75	95,62	95,62	95,75	95,75	95,51	95,22
<b>Eficiencia Ponderada (%)</b>		96,75	97,10	97,10	97,10	97,10	97,10	96,96	96,96

**Tabla 6.2. Número de barras obtenidas por los criterios Grasp Construcción para el Grupo 1.**

En la Tabla 6.2 se muestran los resultados de las instancias del Grupo 1, agrupando los criterios de GRASP construcción sin proceso de mejoría y el otro grupo de criterios GRASP construcción con proceso de mejoría. Los resultados numéricos de cada instancia de prueba, muestran el número de barras utilizadas por cada criterio GRASP construcción y el número óptimo de barras que es la solución óptima.

Además se calcula la Eficiencia Promedio y la Eficiencia Ponderada, donde la Eficiencia Ponderada muestra un resultado relacionado con el resultado óptimo y el resultado de los criterios GRASP construcción de cada instancia.

Las eficiencias se calculan de la siguiente manera:

$$Eficiencia Promedio = \frac{\sum \left( 100 - \left( \frac{x^H - x^O}{x^O} \right) \times 100 \right)}{N^{\circ} Instancias},$$

$$Eficiencia Ponderada = 100 - \left( \frac{\sum x^H - \sum x^O}{\sum x^O} \right) \times 100$$

Donde:

$x^H$  = Número de barras del criterio GRASP construcción por cada Instancia

$x^O$  = Número óptimo de barras por cada Instancia

De forma análoga, se calcula las eficiencias para los demás Grupos de instancias.

CODIGO	OPTIMO	GRASP CONSTRUCCION				GRASP CONSTRUCCION + MEJORIA			
		FFD	BFD	FFDR	BFDR	FFD	BFD	FFDR	BFDR
ME-004	330	352	351	344	344	351	351	343	343
ME-003	348	357	357	357	357	357	357	357	357
ME-011	362	396	368	368	394	394	394	367	368
ME-010	683	709	708	708	708	708	708	707	708
ME-001	671	680	680	676	676	680	677	677	676
ME-019	362	396	395	368	368	395	394	367	367
Eficiencia Promedio (%)		94,47	95,88	97,58	96,38	94,68	94,80	97,72	97,67
Eficiencia Ponderada (%)		95,14	96,26	97,64	96,70	95,32	95,46	97,75	97,71

Tabla 6.3. Número de barras obtenidas por los criterios Grasp Construcción para el Grupo 2.

CODIGO	OPTIMO	GRASP CONSTRUCCION				GRASP CONSTRUCCION + MEJORIA			
		FFD	BFD	FFDR	BFDR	FFD	BFD	FFDR	BFDR
ME-010	942	972	991	991	991	991	991	990	990
ME-050	124	129	128	128	128	128	128	128	128
Eficiencia Promedio (%)		96,39	95,79	95,79	95,79	95,79	95,79	95,84	95,84
Eficiencia Ponderada (%)		96,72	95,03	95,03	95,03	95,03	95,03	95,12	95,12

Tabla 6.4. Número de barras obtenidas por los criterios Grasp Construcción para el Grupo 3.

CODIGO	OPTIMO	GRASP CONSTRUCCION				GRASP CONSTRUCCION + MEJORIA			
		FFD	BFD	FFDR	BFDR	FFD	BFD	FFDR	BFDR
P-001	74	76	76	76	76	76	76	76	76
P-002	90	93	93	93	93	93	93	93	93
P-003	42	43	43	43	43	43	43	43	43
P-004	22	22	22	22	22	22	22	22	22
P-005	340	344	344	352	342	342	342	342	342

P-006	23	23	23	23	23	23	23	23	23
P-007	22	23	23	23	23	23	23	23	23
P-008	51	55	55	55	55	55	55	55	55
P-009	68	68	68	68	68	68	68	68	68
P-010	21	21	21	21	21	21	21	21	21
P-011	503	506	505	505	505	505	505	504	504
P-012	20	20	20	20	20	20	20	20	20
P-013	33	35	35	34	34	35	35	34	34
P-014	20	21	21	21	21	21	21	21	21
P-015	86	87	86	86	86	86	86	86	86
P-016	60	62	62	62	62	62	62	62	62
P-017	103	107	107	106	106	107	107	106	106
P-018	567	598	598	598	598	598	598	597	598
P-019	38	40	40	40	40	40	40	40	39
P-020	70	71	71	70	70	70	71	70	70
P-021	48	49	49	49	49	49	49	49	49
P-022	203	204	204	204	204	204	204	204	204
P-023	106	109	109	108	108	109	109	108	108
P-024	75	77	77	76	76	77	77	76	76
P-025	178	186	186	186	186	186	186	186	186
P-026	118	125	125	125	125	125	125	125	125
P-027	82	88	88	88	88	88	88	88	88
P-028	184	200	198	199	199	198	198	198	199
P-029	270	272	272	271	271	271	272	271	271
P-030	392	402	401	397	397	401	401	397	397
P-031	111	115	115	115	115	115	115	114	115
P-032	50	53	52	52	52	52	52	52	52
P-033	125	129	129	129	129	129	129	129	129
P-034	74	75	75	74	74	75	75	74	74
P-035	65	69	69	69	69	69	69	69	69
<b>Eficiencia Promedio (%)</b>		96,79	96,93	97,14	97,23	96,99	96,94	97,28	97,31
<b>Eficiencia Ponderada (%)</b>		96,91	97,05	97,09	97,32	97,14	97,09	97,42	97,37

Tabla 6.5. Número de barras obtenidas por los criterios Grasp Construcción para el Grupo 4.

CODIGO	OPTIMO	GRASP CONSTRUCCION				GRASP CONSTRUCCION + MEJORIA			
		FFD	BFD	FFDR	BFDR	FFD	BFD	FFDR	BFDR
113816610	351	356	356	356	356	356	356	356	356
717876640	10	10	10	10	11	10	10	10	10
115084640	6	6	6	6	6	6	6	6	6
717871641	5	5	5	5	5	5	5	5	5
119082640	4	4	4	4	4	4	4	4	4
119246640	2	2	2	2	2	2	2	2	2
130808640	2	2	2	2	2	2	2	2	2
130810640	3	4	3	3	3	3	3	3	3
130813610	629	637	637	637	637	637	637	637	637
130813640	144	146	146	146	146	146	146	146	146
130814640	13	13	13	13	13	13	13	13	13
130816810	9	9	9	9	9	9	9	9	9
130817640	20	20	20	20	20	20	20	20	20
130819640	3	3	3	3	3	3	3	3	3
130820640	2	2	2	2	2	2	2	2	2
731606640	6	6	6	6	6	6	6	6	6
717834640	9	9	9	9	9	9	9	9	9
712644640	3	4	3	3	3	3	3	3	3
717832610	2	2	2	2	2	2	2	2	2
717833640	9	10	10	10	10	10	10	10	10
717837810	400	467	466	466	466	466	466	466	466
717844640	63	64	64	64	64	64	64	64	64
<b>Eficiencia Promedio (%)</b>		95,45	98,49	98,49	98,03	98,49	98,49	98,49	98,49
<b>Eficiencia Ponderada (%)</b>		94,93	95,10	95,10	95,04	95,10	95,10	95,10	95,10

Tabla 6.6. Número de barras obtenidas por los criterios Grasp Construcción para el Grupo 5 STADLER.

CODIGO	OPTIMO	GRASP CONSTRUCCION				GRASP CONSTRUCCION + MEJORIA			
		FFD	BFD	FFDR	BFDR	FFD	BFD	FFDR	BFDR
ac01	124	131	130	130	130	130	130	130	130
ac02	14	15	15	15	15	15	15	15	15
<b>Eficiencia Promedio (%)</b>		93,61	94,01	94,01	94,01	94,01	94,01	94,01	94,01
<b>Eficiencia Ponderada (%)</b>		97,92	97,92	97,92	97,92	97,92	97,92	97,92	97,92

Tabla 6.7. Número de barras obtenidas por los criterios Grasp Construcción para el Grupo 6 PAUTA.

CODIGO	OPTIMO	GRASP CONSTRUCCION				GRASP CONSTRUCCION + MEJORIA			
		FFD	BFD	FFDR	BFDR	FFD	BFD	FFDR	BFDR
Ferro1	38	39	39	39	39	39	39	39	39
Ferro2	10	10	10	10	10	10	10	10	10
<b>Eficiencia Promedio (%)</b>		98,68	98,68	98,68	98,68	98,68	98,68	98,68	98,68
<b>Eficiencia Ponderada (%)</b>		97,92	97,92	97,92	97,92	97,92	97,92	97,92	97,92

**Tabla 6.8. Número de barras obtenidas por los criterios Grasp Construcción para el Grupo 7 DEGRAEVE.**

CODIGO	OPTIMO	GRASP CONSTRUCCION				GRASP CONSTRUCCION + MEJORIA			
		FFD	BFD	FFDR	BFDR	FFD	BFD	FFDR	BFDR
goulim1	51	55	54	53	53	54	54	53	53
goulim2	59	62	61	61	61	61	62	61	61
<b>Eficiencia Promedio (%)</b>		93,54	95,36	96,34	96,34	95,36	94,52	96,34	96,34
<b>Eficiencia Ponderada (%)</b>		93,64	95,45	96,36	96,36	95,45	94,55	96,36	96,36

**Tabla. 6.9 Número de barras obtenidas por los criterios Grasp Construcción para el Grupo 8 GOULIMIS.**

CODIGO	OPTIMO	GRASP CONSTRUCCION				GRASP CONSTRUCCION + MEJORIA			
		FFD	BFD	FFDR	BFDR	FFD	BFD	FFDR	BFDR
p-001a	8	9	9	9	9	9	9	9	9
p-002a	23	23	23	23	23	23	23	23	23
p-003a	15	15	15	16	16	15	15	16	15
p-004a	19	19	19	19	19	19	19	19	19
<b>Eficiencia Promedio (%)</b>		96,88	96,88	95,21	95,21	96,88	96,88	95,21	96,88
<b>Eficiencia Ponderada (%)</b>		98,46	98,46	96,92	96,92	98,46	98,46	96,92	98,46

**Tabla 6.10. Número de barras obtenidas por los criterios Grasp Construcción para el Grupo 9 HINTERDING & KHAN.**



CODIGO	OPTIMO	GRASP CONSTRUCCION				GRASP CONSTRUCCION + MEJORIA			
		FFD	BFD	FFDR	BFDR	FFD	BFD	FFDR	BFDR
P-001	2	2	2	2	2	2	2	2	2
P-002	2	2	2	2	2	2	2	2	2
P-003	2	2	2	2	2	2	2	2	2
P-004	2	2	2	2	2	2	2	2	2
P-005	2	2	3	2	2	2	3	2	2
P-006	2	2	2	2	2	2	2	2	2
P-007	2	2	2	2	2	2	2	2	2
P-008	2	3	2	2	2	2	2	2	2
P-009	2	2	2	2	2	2	2	2	2
P-010	2	2	2	2	2	2	2	2	2
P-011	2	2	2	2	2	2	2	2	2
P-012	2	2	2	2	2	2	2	2	2
P-013	2	2	2	2	2	2	2	2	2
P-014	2	2	2	2	2	2	2	2	2
P-015	3	3	3	3	3	3	3	3	3
<b>Eficiencia Promedio (%)</b>		96,67	96,67	100,00	100,00	100,00	96,67	100,00	100,00
<b>Eficiencia Ponderada (%)</b>		96,77	96,77	100,00	100,00	100,00	96,77	100,00	100,00

**Tabla 6.11. Número de barras obtenidas por los criterios Grasp Construcción para el Grupo 10 ALOISE & MACULAN.**

## 6.4 Análisis de Resultados

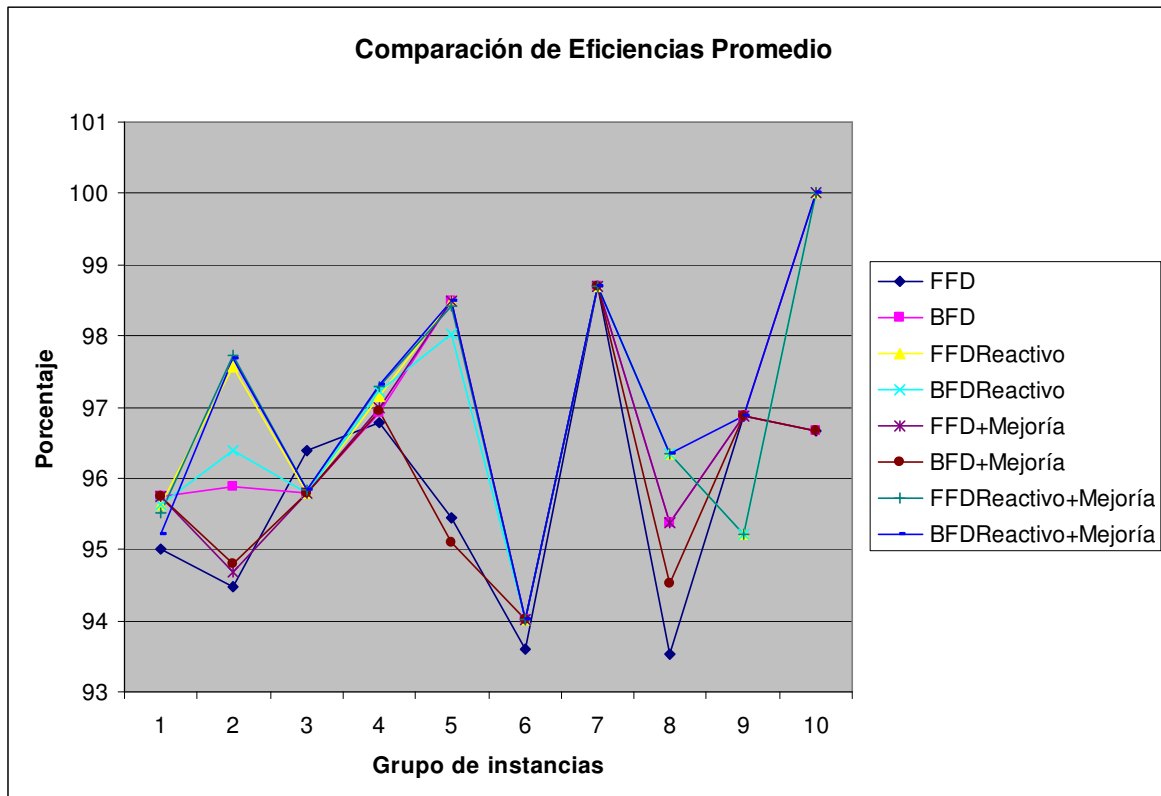
### 6.4.1 Eficiencia Promedio

La Tabla 6.12 muestra las eficiencias promedio por grupo de instancias, por criterio GRASP Construcción y por el número de barras utilizadas.

Grupo de Instancias	GRASP CONSTRUCCION				GRASP CONSTRUCCION + MEJORIA			
	FFD	BFD	FFDR	BFDR	FFD	BFD	FFDR	BFDR
Grupo 1	95,00	95,75	95,62	95,62	95,75	95,75	95,51	95,22
Grupo 2	94,47	95,88	97,58	96,38	94,68	94,80	97,72	97,67
Grupo 3	96,39	95,79	95,79	95,79	95,79	95,79	95,84	95,84
Grupo 4	96,79	96,93	97,14	97,23	96,99	96,94	97,28	97,31
Grupo 5 STADLER	95,45	98,49	98,49	98,03	98,49	98,49	98,49	98,49
Grupo 6 PAUTA	93,61	94,01	94,01	94,01	94,01	94,01	94,01	94,01
Grupo 7 DEGRAEVE	98,68	98,68	98,68	98,68	98,68	98,68	98,68	98,68
Grupo 8 GOULIMIS	93,54	95,36	96,34	96,34	95,36	94,52	96,34	96,34
Grupo 9 HINTERDING & KHAN	96,88	96,88	95,21	95,21	96,88	96,88	95,21	96,88
Grupo 10 ALOISE & MACULAN	96,67	96,67	100,00	100,00	100,00	96,67	100,00	100,00
<b>Promedio Eficiencia Promedio (%)</b>	95,75	96,44	96,89	96,73	96,66	95,91	96,90	97,04

**Tabla 6.12. Tabla resumen de eficiencias promedio.**

En la Tabla 6.12, que es el resumen de los resultados obtenidos, se observa que para el GRASP construcción sin proceso de mejoría, los criterios FFDR y BFDR aportan mejores resultados que los criterios FFD y BFD; también se observa que para el GRASP construcción con proceso de mejoría, los criterios FFDR y BFDR obtienen mejores resultados que los criterios FFD y BFD. Además se supera la eficiencia promedio obtenido por el GRASP básico en [Ganoza+02] cuya eficiencia es de 95,75%.



**Figura 6.1. Gráfica del promedio de la mejor eficiencia de GRASP Reactivo.**

En la Figura 6.1 se muestra que el algoritmo GRASP Reactivo posee mejor eficiencia promedio que el algoritmo GRASP básico.

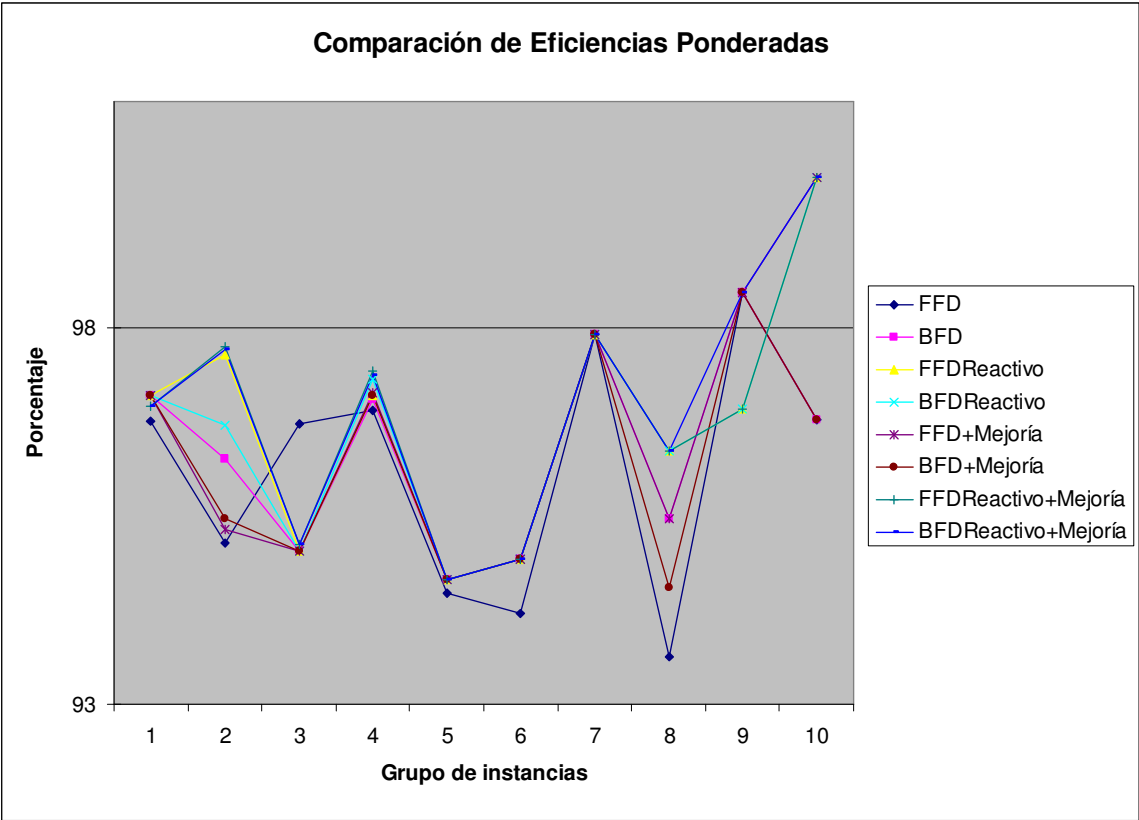
### 6.4.2 Eficiencia Ponderada

La Tabla 6.13 muestra las eficiencias ponderadas por grupo de instancias, por criterio GRASP Construcción y por el número de barras utilizadas.

Grupo de Instancias	GRASP CONSTRUCCION				GRASP CONSTRUCCION + MEJORIA			
	FFD	BFD	FFDR	BFDR	FFD	BFD	FFDR	BFDR
Grupo 1	96,75	97,10	97,10	97,10	97,10	97,10	96,96	96,96
Grupo 2	95,14	96,26	97,64	96,70	95,32	95,46	97,75	97,71
Grupo 3	96,72	95,03	95,03	95,03	95,03	95,03	95,12	95,12
Grupo 4	96,91	97,05	97,09	97,32	97,14	97,09	97,42	97,37
Grupo 5 STADLER	94,93	95,10	95,10	95,04	95,10	95,10	95,10	95,10
Grupo 6 PAUTA	94,20	94,93	94,93	94,93	94,93	94,93	94,93	94,93
Grupo 7 DEGRAEVE	97,92	97,92	97,92	97,92	97,92	97,92	97,92	97,92
Grupo 8 GOULIMIS	93,64	95,45	96,36	96,36	95,45	94,55	96,36	96,36
Grupo 9 HINTERDING & KHAN	98,46	98,46	96,92	96,92	98,46	98,46	96,92	98,46
Grupo 10 ALOISE & MACULAN	96,77	96,77	100,00	100,00	100,00	96,77	100,00	100,00
<b>Ponderado Eficiencia Ponderada ( %)</b>	96,14	96,41	96,81	96,73	96,65	96,20	97,19	96,99

**Tabla 6.13. Tabla resumen de eficiencias ponderadas.**

En la Tabla 6.13, que es el resumen de los resultados obtenidos de eficiencias ponderadas, se observa que para el GRASP construcción sin proceso de mejoría, los criterios FFDR y BFDR aportan mejores resultados que los criterios FFD y BFD; también se observa que para el GRASP construcción con proceso de mejoría, los criterios FFDR y BFDR obtienen mejores resultados que los criterios FFD y BFD. Por lo tanto el GRASP Reactivo obtiene mejores resultados que los criterios GRASP FFD y BFD.



**Figura 6.2. Gráfica del ponderado de la mejor eficiencia de GRASP Reactivo.**

En la Figura 6.2 se muestra que el algoritmo GRASP Reactivo posee mejor eficiencia ponderada que el algoritmo GRASP básico.

### 6.4.3 Convergencia

La Tabla 6.14 muestra el promedio del número de iteraciones que fueron necesarias para encontrar una solución, por grupo, por variante GRASP, por criterio de Construcción y por el número de iteraciones.

Grupo de Instancias	GRASP CONSTRUCCION				GRASP CONSTRUCCION + MEJORIA			
	FFD	BFD	FFDR	BFDR	FFD	BFD	FFDR	BFDR
Grupo 1	2000	2000	2000	2000	2000	2000	2000	2000
Grupo 2	2000	2000	2000	2000	2000	2000	2000	2000
Grupo 3	2000	2000	2000	2000	2000	2000	2000	2000
Grupo 4	1726,66	1664,03	1772,97	1714,94	1715,63	1686,77	1718,11	1601,31
Grupo 5 STADLER	546,23	476,95	637,05	546,32	546,18	546,18	546,18	509,41
Grupo 6 PAUTA	2000	2000	2000	2000	2000	2000	2000	2000
Grupo 7 DEGRAEVE	1000,50	1001,00	1001,00	1000,50	1000,50	2	1000,50	1000,50
Grupo 8 GOULIMIS	2000	2000	2000	2000	2000	2000	2000	2000
Grupo 9 HINTERDING & KHAN	4,75	12,25	5,00	8,50	8,00	2,00	9,75	1,50
Grupo 10 ALOISE & MACULAN	1,87	134,27	1,13	1,00	1,60	134,60	1,13	1,00
<b>Promedio N°iteraciones</b>	1328,00	1328,85	1341,72	1327,13	1327,19	1237,16	1327,57	1311,37

**Tabla 6.14. Tabla resumen de convergencia.**

En la Tabla 6.14, que es el resumen de los resultados obtenidos, en el peor de los casos la convergencia promedio se da en 1341.72 iteraciones para el caso del GRASP Reactivo FFD sin proceso de Mejoría y, en el mejor de los casos, la convergencia promedio se da en 1237.16 iteraciones para el GRASP con criterio BFD con proceso de Mejoría.

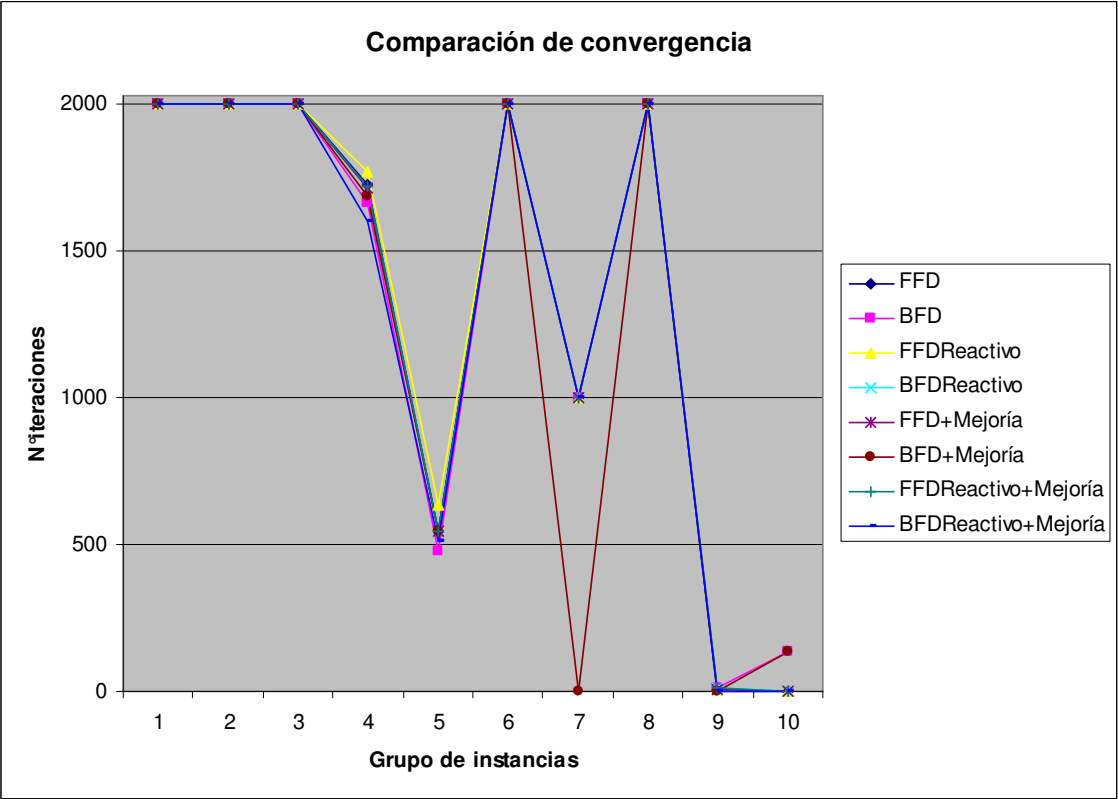


Figura 6.3. Gráfica de la mejor convergencia de GRASP.

## Capítulo 7: Conclusiones y trabajos futuros

En el problema de cortes 1D es muy importante dar una solución apropiada para reducir las pérdidas de desperdicio en diversas industrias tales como la del papel, construcción, Maderera, vidrio y aluminio, donde se manejan materias primas de dimensión lineal que requieren ser cortadas en dimensiones menores a una estándar, . Por ello, es necesario resolverlo de manera óptima mediante algoritmos exactos o cercanos a la óptima mediante algoritmos meta heurísticos.

Se observó que el algoritmo GRASP provee una forma sencilla de implementar y obtener mejores resultados a comparación de otras meta heurísticas descritas, así mismo, ofrece flexibilidad y robustez para implementarla en algún tipo de problema específico.

El software provee opción de ejecución para el GRASP básico y el GRASP Reactivo en forma paralela. Además permite la carga automática de datos en formato Excel.

La variante GRASP Reactiva aporta resultados mejores a las del algoritmo GRASP básico debido a que su parámetro de relajación se ajusta de manera automática conforme encuentra una mejor solución.

El GRASP Reactivo mejora los resultados de eficiencia promedio obtenidos con el GRASP básico propuesto por Ganoza [Ganoza+02] en 1,19%.

El GRASP Reactivo con criterio BFD con proceso de mejoría aporta resultados favorables mediante una eficiencia promedio obteniendo 97,04% frente a un 96,90% para el caso del GRASP Reactivo BFD sin proceso de Mejoría.

El GRASP Reactivo con criterio BFD aporta resultados favorables mediante una eficiencia ponderada obteniendo 97,19% frente a un 96,73% para el caso del GRASP Reactivo BFD sin proceso de Mejoría.

Se obtuvo una convergencia promedio menor para el GRASP BFD con proceso de Mejoría con 1237,16 iteraciones para encontrar una solución.

La utilización de 2 parámetros de relajación, el primer parámetro para obtener aleatoriamente la pieza a procesar y el segundo parámetro para la selección aleatoria de la barra a cortar, obtiene resultados favorables debido a que se selecciona de un



conjunto de piezas y de un conjunto de barras, aleatoriamente, lo cual diversifica las posibles soluciones.

Como trabajo futuro, se considera agregar la cantidad de desperdicio que genere la cuchilla de corte.

Para mejorar el procesamiento de requerimientos de una misma longitud y cantidades enormes de ellas se deberá aplicar una agrupación para la repetición de patrones.

El problema de cortes 1D posee su variante para el corte de barras de distintas dimensiones, el cual se debe seleccionar la barra que genere menor desperdicio.

Se puede complementar al GRASP Construcción Reactivo técnicas alternativas tales como las variantes GRASP con bias y Path Relinking.

# **Anexos**

## ANEXO A: Instancias de Prueba

### Grupo #1: Nacif Rocha [Nacif 97]

Tamaño de las Barras: 6000mm

Instancia	Tamaño	Cantidad
20-347	3853	4
	3733	2
	2853	4
	2773	10
	2753	2
	2200	10
	2100	4
	2053	76
	1853	5
	1353	12
	1253	436
	1053	218
	653	218
	553	76
ME-28	1962	176
	1462	16
	980	56
	818	428
	805	24
	595	32
	529	428
	381	192
ME-29	1162	48
	1062	32
	980	56
	931	152
	930	16
	895	16
	865	8
	818	428
	805	40
	931	16
	625	40
	595	32
	529	428
	462	436
	362	152

Instancia	Tamaño	Cantidad
20-348	2153	36
	1853	15
	1200	10
	1053	76
	953	5
	828	50
	753	28
	653	28
	553	76
20-349	3003	5
	2953	4
	2903	3
	2853	2
	2803	1
	2753	13
	2553	20
	2450	10
	2400	8
	2350	6
	2300	4
	2250	2
	2200	32
	1853	13
	1653	8
	1253	516
	1053	10
	953	116
CM-063	2528	20
	1828	10
	1628	8
	1228	302
	1227	222
	1027	10
	927	116
	628	8

Instancia	Tamaño	Cantidad
CM-096	2978	5
	2928	4
	2878	3
	2828	2
	2778	1
	2728	13
	2477	10
	2427	8
	2377	6
	2327	4
	2277	2
	2227	32
	1828	3
ME-001	826	20
	800	6
	726	16
	653	20
	640	16
	628	12
	615	8
	603	4
	590	52
	565	80
	526	176
	524	428
ME-011	2910	5
	2860	4
	2810	3
	2760	2
	2710	1
	2660	13
	2640	10
	1760	8
	1560	4
	1160	44

Instancia	Tamaño	Cantidad
CM-030	1105	80
	1080	64
	1055	48
	1030	32
	1005	16
	980	232
	862	20
	831	20
	818	428
	805	24
	762	272
	731	16
	658	80
	645	64
	633	48
	620	32
	608	16
	595	208
	570	80
	531	176
	529	428

**Grupo #2: Nacif Rocha [Nacif 97]**

Tamaño de las Barras: 6000mm

Instancia	Tamaño	Cantidad
ME-001	801	80
	776	920
	751	720
	726	120
	706	240
	701	2048
	676	1104
	626	40
	576	40
	526	40
	476	40
	426	40
	416	40
	411	40
	406	40
	401	40
	376	40
	351	40
	346	40
	341	40
	336	40
	331	40
	326	40
ME-003	1653	20
	1603	230
	1553	180
	1503	30
	1463	60
	1453	512
	1403	276
	1303	10
	1203	10
	1103	10
	1003	10
	903	10
	883	10
	873	10
	863	10
	853	10
	803	10
	753	10
	743	10
	733	10
	723	10
	713	10
	703	10

Instancia	Tamaño	Cantidad
ME-004	1552	20
	1512	230
	1502	180
	1402	20
	1392	60
	1362	512
	1352	10
	1302	256
	1252	20
	1152	20
	1102	10
	1052	20
	932	10
	922	10
	912	10
	902	10
	852	10
	802	10
	792	10
	782	10
	772	10
	762	10
	752	10
ME-010	1600	40
	1560	460
	1550	360
	1450	40
	1440	120
	1410	1024
	1400	20
	1350	512
	1300	40
	1200	40
	1150	20
	1100	40
	980	20
	970	20
	960	20
	950	20
	900	20
	850	20
	840	20
	830	20
	820	20
	810	20
	800	20

ANEXO A: Instancias de Prueba

Instancia	Tamaño	Cantidad
ME-011	1710	20
	1660	230
	1610	180
	1560	30
	1520	60
	1510	512
	1460	276
	1360	10
	1260	10
	1160	10
	1060	10
	960	10
	940	10
	930	10
	920	10
	910	10
	860	10
	810	10
	800	10
	790	10
	780	10
	770	10
	760	10
ME-019	1710	20
	1660	230
	1610	180
	1560	30
	1520	60
	1510	512
	1460	276
	1360	10
	1260	10
	1160	10
	1060	10
	960	10
	940	10
	930	10
	920	10
	910	10
	860	10
	810	10
	800	10
	790	10
	780	10
	770	10
	760	10

**Grupo # 3: Nacif Rocha [Nacif97]**

Tamaño de las Barras: 6000mm

Instancia	Tamaño	Cantidad
ME-050	998	20
	993	20
	988	20
	983	20
	978	20
	976	20
	973	20
	968	20
	963	20
	962	75
	960	70
	958	65
	957	20
	956	60
	954	55
	952	50
	882	45
	872	40
	862	35
	852	30
	767	25
	762	20
	757	15
	752	10
ME-050	1600	40
	1560	460
	1550	360
	1450	40
	1440	120
	1410	1024
	1400	20
	1650	512
	1300	40
	1200	40
	1150	20
	1100	40
	1046	40
	1041	40
	1036	40
	1031	40
	1026	40
	1024	40
	1021	40
	1016	40
	1011	40
	1010	150

Instancia	Tamaño	Cantidad
	1008	140
	1006	130
	1005	40
	1004	120
	1002	110
	1000	100
	980	20
	970	20
	960	20
	950	20
	930	90
	920	80
	910	70
	900	80
	850	20
	840	20
	830	20
	820	20
	815	50
	810	60
	805	30
	800	40

**Grupo # 4: Nacif Rocha [Nacif97]**

Tamaño de las Barras: 6000mm a 6100mm

Instancia	Tamaño	Cantidad
P-001	1950	8
	1850	4
	1450	230
	1250	32
	1150	32
	550	17
P-002	1500	64
	1439	32
	1433	12
	1431	64
	950	32
	948	54
	939	96
	931	96
	689	2
	683	4
	681	2
	533	30
	433	2
	333	2
P-003	2974	54
	1874	16
	1474	2
	1174	45
	974	1
	774	1
P-004	2000	8
	1900	4
	1500	4
	1300	32
	1200	32
	600	32
P-005	6050	32
	3050	94
	2520	34
	2120	2
	2050	172
	2020	2
	1950	36
	1550	144
	1350	32
	1250	284
	1050	120

Instancia	Tamaño	Cantidad
	950	64
	850	2
	650	189
	550	172
	450	98
P-006	2000	2
	1200	64
	900	32
	600	26
	400	32
P-007	1950	4
	1850	2
	1450	50
	1250	16
	1150	16
	550	16
P-008	2456	34
	2056	2
	1950	4
	1850	2
	1450	115
	1250	16
	1150	16
	550	1
P-009	950	4
	941	64
	844	32
	800	64
	747	32
	600	32
	550	96
	544	26
	541	134
	453	14
	441	136
	344	32
P-010	914	4
	808	32
	600	32
	514	96
	508	26
	308	32



ANEXO A: Instancias de Prueba

Instancia	Tamaño	Cantidad
P-011	1848	16
	1748	8
	1500	4
	1454	128
	1439	32
	1433	24
	1432	128
	1431	160
	1348	460
	1274	102
	1048	64
	1000	55
	948	216
	939	96
	931	300
	914	4
	904	64
	882	64
	881	64
	874	6
	808	32
	800	64
	750	64
	689	2
	687	32
	683	8
	681	4
	600	32
	533	60
	514	96
	511	30
	508	26
	500	47
	481	134
	472	45
	448	4
	435	8
	433	4
	419	192
	393	14
	382	64
	381	136
	333	4
	308	32
	300	64
	235	192
	232	52
	132	64

Instancia	Tamaño	Cantidad
P-012	860	32
	600	64
	560	14
	436	14
	400	64
	337	32
P-013	1200	18
	1120	45
	1050	73
	1000	36
	872	2
	560	10
	525	6
	426	4
P-014	2100	18
	2000	10
	1200	18
	1120	15
	1050	14
	1000	4
P-015	2000	4
	1200	192
	1000	46
	900	64
	600	186
	500	68
P-016	400	64
	2974	64
	1874	32
	1500	4
	1000	38
	500	38
P-017	483	96
	1439	32
	1433	12
	1432	64
	1431	64
	1000	17
	948	54
	939	96
	931	96
	882	32
	689	2

ANEXO A: Instancias de Prueba

Instancia	Tamaño	Cantidad
	683	4
	681	2
	542	60
	535	60
	533	30
	500	9
	433	2
	333	2
P-018	1439	704
	1427	176
	1000	119
	939	2112
	689	44
P-019	3500	16
	1974	40
	1000	64
	761	30
	738	3
P-020	674	3
	654	80
	641	256
	544	6
	541	160
	450	128
	224	220
P-021	901	160
	662	60
	618	80
	414	128
P-022	1048	268
	933	24
	932	148
	919	12
	901	80
	875	12
	693	120
	662	30
	639	3
	618	80
	581	256
	484	6
	481	160
	445	160

Instancia	Tamaño	Cantidad
	414	128
	326	120
	297	6
	287	160
	185	256
	164	220
P-023	3500	34
	2974	30
	1974	80
	1000	128
	700	128
	600	34
	283	110
P-024	2975	4
	1675	64
	1475	181
	1175	5
	973	56
P-025	2250	128
	2185	4
	2150	120
	1800	112
	1200	256
P-026	2210	64
	2145	2
	2110	60
	1165	112
	1161	12
	1158	248
P-027	2210	64
	2145	2
	2110	60
	1165	56
	1161	8
	1158	124
P-028	1069	8
	876	8
	784	672
	765	672
	687	40
	527	16
	253	40

## ANEXO A: Instancias de Prueba

Instancia	Tamaño	Cantidad
P-029	1065	32
	1062	372
	1044	8
	806	8
	716	32
	706	484
	668	64
	628	128
	556	4
	554	120
	544	112
	530	12
	518	253
	516	248
	501	8
	489	120
	478	250
	476	12
	312	128
	243	12
	237	250
P-030	1148	256
	1132	2
	1083	8
	1069	224
	1048	240
	856	504
	809	496
	709	480
	559	32
	554	258
	532	32
	457	224
	452	224
P-031	809	124
	806	8
	716	32
	709	120
	706	484
	559	8
	556	4
	457	224

Instancia	Tamaño	Cantidad
P-032	688	128
	561	8
	549	120
	538	262
P-033	2300	8
	1200	114
	1154	6
	750	128
	704	128
	600	288
	554	404
P-034	1675	64
	1475	181
	1175	5
	973	56
P-035	1975	1
	1775	4
	1675	65
	1475	180
	1175	4

**Grupo STADTLER: [Stadtler90]**

Tamaño de las Barras: 6005mm

Instancia	Tamaño	Cantidad
113816610	6000	82
	4750	21
	4500	223
	3285	3
	2600	32
	2250	25
717876640	2045	1
	2026	4
	1988	1
	1926	4
	1826	4
	1726	4
	1626	4
	1526	4
	1426	5
	1326	3
115084640	2135	6
	1385	2
	1380	2
	1260	10
	885	2
	750	2
717871641	2064	6
	2029	2
	1947	2
	947	2
	929	4
	910	1
	900	1
119082640	2500	2
	2000	4
	1125	1
	1000	2
	875	2
	760	1
119246640	1135	5
	1010	1
	510	1
130808640	965	4
	894	4

Instancia	Tamaño	Cantidad
130810640	2086	4
	1903	4
130813610	6000	156
	4602	32
	4352	412
	2452	48
	2102	26
	130	674
130813640	2394	16
	2194	16
	1996	2
	1994	16
	1985	2
	1956	4
	1947	2
	1938	6
	1738	6
	1538	6
	1447	2
	1338	6
	1138	6
	938	8
	928	2
	912	2
	900	2
	885	2
	813	2
	738	6
	554	360
	553	340
	538	6
	492	10
	421	10
	373	704
	312	16
132068610	6000	82
	4750	21
	4500	223
	3285	3
	2600	32
	2250	25

## ANEXO A: Instancias de Prueba

Instancia	Tamaño	Cantidad
130814640	2435	2
	2029	4
	2020	2
	2013	4
	1991	4
	1274	2
	963	2
	962	2
	956	4
	928	4
	910	2
	572	24
	407	24
130816810	1352	20
	1196	20
130817640	2029	8
	2013	6
	1279	4
	1206	4
	1154	4
	1057	10
	1013	12
	963	2
	962	2
	956	12
	938	4
	932	4
	779	2
	698	2
	654	2
	644	4
	450	12
	448	2
	325	24
	274	35
130819640	1854	2
	1814	4
	1305	2
	786	2
	671	2
130820640	1991	4
	928	4

Instancia	Tamaño	Cantidad
731606640	2040	2
	2035	4
	2000	4
	1447	2
	928	6
	813	2
13207040	1135	5
	1010	1
	510	1
712644640	2130	4
	1903	4
717832610	1925	2
	1539	2
	1000	2
717833640	4011	2
	2135	6
	1135	22
	1010	5
717837810	2400	400
	1750	800

**Grupo PAUTA: [Mota89]**

Tamaño de las Barras: 12000mm para AC-01 y 11500mm para AC-02.

Instancia	Tamaño	Cantidad
AC-01	6770	11
	6050	32
	5970	32
	5500	32
	5470	32
	4920	12
	4700	16
	4400	16
	4000	50
	3550	32
	3350	32
	2600	17
AC-02	7300	5
	6900	2
	6400	2
	5800	4
	5400	2
	4900	2
	3400	4
	2250	4
	2050	3
	1870	4
	1700	5
	1500	2
	1370	3

**Grupo DEGRAEVE: [Degraeve+97]**

Tamaño de las Barras: 6000mm para Ferror1 y 3000mm para Ferro2

Instancia	Tamaño	Cantidad
Ferro1	1910	15
	1850	15
	1790	30
	1110	15
	1101	45
	550	15
	500	75
	340	15
Ferro2	560	15
	491	15
	390	30

**Grupo GOULIMIS: [Goulimis90]**

Tamaño de las Barras: 4300mm

Instancia	Tamaño	Cantidad
Goulim01	2350	2
	2250	4
	2200	4
	2100	15
	2050	6
	2000	11
	1950	6
	1900	15
	1850	13
	1700	5
	1650	2
	1350	9
	1300	3
	1250	6
	1200	10
	1150	4
	1100	8
	1050	3

Instancia	Tamaño	Cantidad
Goulim02	2200	24
	2150	4
	2100	12
	2050	18
	2000	2
	1950	7
	1900	4
	1850	2
	1750	11
	1650	4
	1550	11
	1450	2
	1300	7
	1250	3
	1200	3
	1150	1
	1100	6
	1050	6
	1000	6
	950	7
	900	7
	850	6

**Grupo HINTERDING & KHAN: [Hinterding94]**

Tamaño de las Barras: 140mm para (P-001a) ,150 para (P-002a) y 250 para (P-003a, P-004a)

Instancia	Tamaño	Cantidad
P-001a	100	3
	90	1
	80	2
	70	4
	60	2
	50	1
	40	2
	30	5
P-002a	100	8
	90	5
	80	5
	70	8
	60	7
	50	5
	40	8
	30	4
P-003a	100	6
	90	4
	80	6
	70	15
	60	5
	50	6
	40	12
	30	6
P-004a	120	1
	110	8
	100	6
	90	4
	80	7
	70	15
	60	12
	50	7



**Grupo ALOISE & MACULAN: [Aloise+91]**

Tamaño de Barras: 100mm

Instancia	Tamaño	Cantidad
P-001	57	1
	48	1
	26	2
	23	1
	20	1
P-002	57	1
	45	1
	29	1
	26	2
	17	1
P-003	73	1
	47	1
	31	1
	22	1
	15	1
	12	1
P-004	67	1
	61	1
	22	1
	20	1
	10	1
	9	1
	4	2
	3	1
P-005	71	1
	64	1
	24	1
	12	2
	9	1
	8	1
P-006	70	1
	41	1
	33	1
	26	1
	15	1
	7	1
	6	1
	2	1

Instancia	Tamaño	Cantidad
P-007	61	1
	57	1
	15	2
	13	1
	12	2
	7	1
	4	1
	3	1
	1	1
P-008	84	1
	41	1
	27	1
	20	1
	12	1
	9	1
	7	1
P-009	60	1
	53	1
	15	2
	14	1
	9	2
	6	3
	5	1
	2	1
P-010	74	1
	49	1
	46	1
	9	1
	6	1
	5	2
	2	3
P-011	66	1
	65	1
	26	1
	9	2
	7	1
	6	1
	5	2
	2	1

Instancia	Tamaño	Cantidad
P-012	39	1
	36	1
	34	1
	30	1
	11	1
	10	2
	9	1
	7	1
	6	1
	5	1
	3	1
P-013	53	1
	48	1
	17	1
	9	3
	8	3
	7	2
	6	2
	4	1
	1	1
P-014	46	1
	45	1
	28	1
	27	1
	14	1
	13	1
	10	1
	8	1
	6	1
	3	1
P-015	69	1
	63	1
	43	1
	37	1
	29	1
	28	1
	20	1
	11	1

## Referencias Bibliográficas

- [Adenso+03] Belarmino Adenso-Díaz y Santiago García-Carbajal, An efficient GRASP algorithm for disassembly sequence planning, Tesis ingeniería, Universidad de Oviedo, España, 2003.
- [Adenso96] Adenso Díaz, B. et al.: Optimización heurística y redes neuronales en dirección de operaciones e ingeniería, Madrid, Editorial Paraninfo, 1996.
- [Aloise+91] Aloise D., Maculan N., Uma classe de algoritmos aproximativos decrescente para o problema 'bin packing', Revista Brasileira de Computacao, v.6 n.3, 1991, pp.3-12.
- [Alvarez+08] R. Alvarez-Valdes, F. Parreño y J.M. Tamarit, Reactive GRASP for the strip-packing problem, Computers and Operations Research, v. 35 , n. 4, 2008, pp. 1065-1083.
- [Ance98] Ance P. "Branch-and-Price Algorithms for the One-Dimensional Cutting Stock Problem", Computational Optimization and Applications, v. 9, n. 3, 1998.
- [Aiex+05] Renata M. Aiex y Mauricio G. C. Resende, Probability Distribution of Solution Time in GRASP: An Experimental Investigation, INFORMS Journal on Computing, v. 17, n. 2, 2005.
- [Bautista+04] Joaquín Bautista y Jordi Pereira, A GRASP algorithm to solve the unicast set covering problem, Tesis ingeniería, Universitat Politècnica de Catalunya, España 2004.
- [Beltran+07] Jesus David Beltran Cano y Jose Eduardo Calderon, Procedimientos constructivos adaptativos (GRASP) para el problema del empaquetado bidimensional, Tesis ingeniería, Universidad de la Laguna, 2007.
- [Beltrán+05] Jesus David Beltran y Jose Eduardo Calderon, GRASP/VNS hybrid for the Strip Packing Problem, tesis ingeniería, Universidad de La Laguna, España, 2005.
- [Beloy02] G. Belov, G. Scheithauer, A cutting plane algorithm for the one-dimensional cutting stock problem with multiple stock lengths. European Journal of Operational, v. 141, 2002, pp. 274-294.
- [Binato+02] S. Binato and G.C y Oliveira, A Reactive GRASP for transmission network expansion planning, In C.C. Ribeiro and P. Hansen, editors, Essays and surveys in metaheuristics, pp. 81-100. Kluwer Academic Publishers, 2002.
- [Boudia+07] M. Boudia, M. A. O. Louly, A reactive GRASP and path relinking for a combined production-distribution problem, Computers and Operations Research, v. 34, n. 11, 2007.
- [Caballero08] Saúl Isaí Caballero Hernández, Un procedimiento de búsqueda voraz, aleatorizado y adaptativo para el diseño eficiente de territorios de atención comercial con requerimientos de asignación conjunta, Tesis de Maestría, Universidad Autónoma de Nuevo León, Nuevo León, España, 2008.
- [Campelo+94] Campelo, R.E. y Maculan, N. Algoritmos e Heurísticas - Desenvolvimento e Avaliação de Performance, Editora: EDUFF, 1994.

- [Caprara98] Caprara A., Toth P., “Lower Bounds and Algorithms for the 2-constrained Bin Packing Problem”, Tech. Report, DEIS OR-98-7, University of Bologna, Bologna, Italy, 1998.
- [Coffman+96] E.G. Coffman, Jr. M.R. Garey, D.S. Johnson. “Approximation Algorithms for Bin Packing: A survey in Approximation Algorithms for NP-Hard Problems”. D. Hochbaum (ed), PWS Publishing Boston, 1996, pp. 46-93.
- [Cruz04] Cruz Reyes, Laura, Clasificación de algoritmos heurísticos para la solución de problemas de bin packing, editora: CENIDET, 2004.
- [Degraeve+97] Degraeve Z., Bemelmans R., “Cutting stock problem at Metal Aarschot”.Departament of Applied Economic Sciences – Katholieke Universiteit Leuven (obtenido de internet).
- [Delgadillo07] Rosa Sumactika Delgadillo Avila, Un estudio algorítmico del problema de corte y empaquetado 2D, Tesis de Licenciatura, Universidad Nacional Mayor de San Marcos , Lima, Perú, 2007.
- [Delorme+03] Xavier Delorme, Xavier Gandibleux y Joaquin Rodriguez, A reactive GRASP and path relinking for a combined production–distribution problem, European Journal of Operational Research 153, 2004, pp. 564–580.
- [Dyckhoff90] Dyckhoff H., “A Typology of Cutting and Packing Problems”, European Journal of Operation Research 44, 1990, pp. 145-159.
- [Dyson76] Dyson R. G., Gregory A. S. “The Cutting Stock Problem in the Flat Glass Industry”, Operational Research Quartely, v. 25, 1976, pp. 41-53.
- [Feo+95] Feo T.A y Resende M.G.C., “Greedy Randomized Adaptive Search Procedures”, Journal of Global Optimization, 1995, pp. 109-133.
- [Foerster+00] Foerster H. y Wascher G., Pattern reduction in one-dimensional cutting stock problems, International Journal of Production Research, v. 38, n. 7, 2000, pp. 1657-1676.
- [Ganoza+02] Ganoza D. y Solano U., “Un algoritmo de búsqueda adaptativa aleatoria y golosa para la resolución del problema de cortes”, Tesis de ingeniería, Universidad Nacional Mayor de San Marcos, Perú, 2002.
- [Gandibleux+98] X. Gandibleux, D. Vancoppenolle y D. Tuytens, A first making use of GRASP for solving MOCO problems, Tesis ingeniería, University of Valenciennes, France, 1998.
- [Garey+79] Garey M. R. y Johnson D. S., Computers and intractability: a guide to the theory of NP-Completeness. New York: W. H. Freeman and Company, 1979.
- [Gau95] T. Gau and G. Wäscher, A problem generator for the standard one-dimensional cutting stock problem, European Journal of Operational Research, v. 84, n. 3, 1995, pp. 572-579.
- [Gilmore+61] P. C. Gilmore y R. E. Gomory. A linear programming approach to the cutting stock problem. Operations Research, 1961, pp. 9:849–859.
- [Glover+03] Glover, F. y Kochenberger, G. A. Handbook of metaheuristics. Kluwer Academic Publishers. Massachusetts, 2003.
- [Glover+03]F. Glover, B. Melián. Búsqueda Tabú. Revista Iberoamericana de Inteligencia Artificial, v. 19, 2003, pp. 29-48.

- [Glover+97] F. Glover and M. Laguna. Tabu Search. Kluwer Academic Publishers, Boston, 1997.
- [Glover89] Glover, F. "Tabú Search: Part I". ORSA Journal on Computing, v. 1, 1989, pp. 190-206.
- [Glover90] Glover, F. "Tabú Search: Part II". ORSA Journal on Computing, v. 2, 1990, pp. 4-32.
- [Gomez+01] Fernando C. Gomes y Panos Pardalos, Reactive GRASP with path relinking for channel assignment in mobile phone networks, Workshop on Discrete Algorithms and Methods for MOBILE Computing and Communications, ISBN:1-58113-421-5, 2001, pp. 60 - 67.
- [Goulimis90] Constantine Goulimis, Optimal solutions for the cutting stock problem, European Journal of Operational, v. 44, n. 2, 1990, pp. 197-208.
- [Hatori98] Hatori S. "One Dimensional Cutting Stock for Plastic Film". Tonen System Plaza Inc. Shibuya - ku, Tokyo 150, Japan .1998.
- [Hernandez04] Hernández C, Diseño de un algoritmo heurístico para el problema de localización p-mediana. tesis de ingeniería, Universidad de las Américas Puebla, Puebla, México,2004.
- [Hinterding94] Hinterding R. "Genetic Algorithms for cutting: an exploration of mapping schemes". Department of Computer and Mathematical Sciences -Victoria University of Technology- Technical Report 40 COMP 12, 1994.
- [Holland73] Holland Jhon, "Genetics Algorithms and the Optimal Allocations of Trials", SIAM Journal of Computing 2(2), 1973, pp. 88-105.
- [Khalifa+06] Y. Khalifa, O. Salem , A. Shahin, Cutting stock waste reduction using genetic algorithms, Proceedings of the 8th annual conference on Genetic and evolutionary computation, ISBN:1-59593-186-4, 2006 , pp. 1675 – 1680.
- [Karelahti02] Janne Karelahti, Solving the cutting stock problem in the steel industry, Tesis de Maestría, HELSINKI UNIVERSITY OF TECHNOLOGY, Helsinki, Finland, 2002.
- [Kantorovich39] Kantorovich, L.V. "Mathematical Methods of Organizing and Planning Production", Management Science 6, 1939, pp. 366-422.
- [Laguna+99] M. Laguna and R. Marti, GRASP and path relinking for 2-layer straight line crossing minimization. INFORMS Journal on Computing, 1999, pp. 11:44-52.
- [Lawler+85] Lawler, E.L., Lenstra, J.K., RinnooyKan, A.H.G. y Shmoys, D.B., TheTravelingSalesmanProblem. A Guided Tour to Combinatorial Optimization, JohnWileyandSons, Chichester, 1985.
- [Leduino+08] Antonio Carlos M. y Luiz Leduino de Salles N. , Nonlinear cutting stock problem model to minimize the number of different patterns and objects, Computational & Applied Mathematics, v. 27, n. 1, 2008.
- [Linderoth+99] Linderoth, J.T. and Savelsbergh, M.W.P., 1999. "A Computational Study of Search Strategies for Mixed Integer Programming". Informs Journal on Computing, 11, 173-187.
- [Marques+00] João P. Marques Silva y Karem A. Sakallah, GRASP—A New Search Algorithm for Satisfiability, ACM, University of Michigan, 2000.

- [Mauricio04] Mauricio David., “Algoritmos GRASP para el problema de cortes de Guillotina”, Reporte técnico FISI, UNMSM.2004.
- [Mauricio+02] Mauricio D. y Delgadillo R., “Algoritmos GRASP para el Problema de Cortes”, Relatorio Técnico N.º 01/2002, Universidad Nacional Mayor de San Marcos, 2002.
- [Morabito+97] Morabito R., Garcia V., “The cutting stock problem in hardboard industry: a case study”, Computer Operation Research, v. 25/, n. 6, 1997, pp 469-485.
- [Mota89] Mota R. “Optimizacao das Perdas em Cortes de Barras para Estructuras de Concreto Armado: Um Sistema Computacional”, Resumos XXII SOBRAPO, 1989.
- [Nacif97] Nacif R, “Optimización de corte de barras”. Tesis aprobada el 05 de febrero de 1997, Belo Horizonte - Brasil.
- [Neto+04] Teresa Neto y João Pedro, GRASP for linear integer programming, Metaheuristics, Applied Optimization, v. 86, 2004, pp. 545-574.
- [Oliveira+07] Oliveira, J. F. y WÄSCHER, G.. Cutting and packing. European Journal of Operacional Research, v. 183, n. 3, 2007, pp. 1106-1108.
- [Osman+96] Osman, I.H. y Kelly, J.P. . Meta-Heuristics: Theory and Applications, Boston USA Ed. Kluwer Academic, 1996.
- [Parreño04] Parreño Torres, Algoritmos heurísticos y exactos para problemas de corte no guillotina en dos dimensiones, Tesis ingeniería, UV, España, 2004.
- [Perez05] Perez F., Una metodología de solución basada en la metaheurística GRASP para el problema de diseño de red con incertidumbre , Tesis de ingeniería, Universidad Autónoma de Nuevo León, Nuevo León, México, 2005.
- [Poldi03] Kelly Cristina Poldi, Algunas extensoes do problema de corte de estoque, Tesis de Maestría, Instituto de Ciencias Matemáticas de Computación , Sao Carlos, Brasil, 2003.
- [Prais+98] M. Prais y C.C. Ribeiro, Reactive GRASP: an application to a matrix decomposition problem in TDMA traffic assignment. INFORMS Journal on Computing, v. 13, n. 3, 2000, pp 164-176.
- [Prais+00] M. Prais y C.C. Ribeiro, Variação de parametros em procedimentos GRASP, Investigación Operativa, 2000, pp. 9:1-20.
- [Resende03] Mauricio G. C. Resende, Procedimientos de búsqueda miopes aleatorizados y adaptativos, Revista Iberoamericana de Inteligencia Artificial, n. 19, 2003, pp. 61-76.
- [Resende+98] Mauricio G. C. Resende, Computing Approximate Solutions of the Maximum Covering Problem with GRASP, Journal of Heuristics, v. 4, n. 2, 1998.
- [Ribeiro+07] Celso C. Ribeiro y Isabel Rosseti, Efficient parallel cooperative implementations of GRASP heuristics, Computer Communications, Universidade Federal Fluminense, Department of Computer Science, 2007.
- [Stadtler90] Hartmut Stadtler, A one-dimensional cutting stock problem in the aluminium industry and its solution, European Journal of Operacional, v. 44, n. 2, 1990, pp. 209-223.

- [Shunji+06] Umetani Shunji, Yagiura Mutsunoti y Ibaraki Toshihide, One-Dimensional Cutting Stock Problem with a Given Number of Setups: A Hybrid Approach of Metaheuristics and Linear Programming, *Journal of Mathematical Modelling and Algorithms*, v. 5, n. 1, 2006, pp. 43-64.
- [Torres03] Torres Sánchez, Algoritmo Genético que resuelve el problema de cortes en una dimensión, Tesis de ingeniería, Pontificia Universidad Católica del Perú, Perú, 2003.
- [Tupia+04] Tupia M. y Mauricio D., “Un algoritmo voraz para resolver el problema de la programación de tareas dependientes en máquinas diferentes”, *Rev. investig. sist. Inform.*, 2004, pp. 9-18.
- [Vieira04] Euclides Vieira Neto, GRASP: Efeito da independência das soluções iniciais na otimização do corte unidimensional, Tesis de Doctorado, Universidade Estadual do Norte Fluminense, Fluminense, Brasil, 2004.
- [Wagner99] Wagner B.J., A genetic algorithm solution for one-dimensional bundled stock cutting, *European Journal of Operational Research*, v. 117, n. 2, 1999, pp. 368-381.
- [Westernlund+95] Westernlund T., Isaksoon J., Harjunkoski I., “Solving a production optimization problem in the paper industry”, Report 95-146A, Process Design Laboratory, Abo Akademi University, 1995.
- [Zabaleta04] Cobos Zabaleta, Nadia. Universidad Autónoma de Nuevo León. Búsqueda Tabú para un problema de diseño de red multiproducto con capacidad finita en las aristas. pp. 6-11. Nuevo León, 2004.